

AD-A188 205

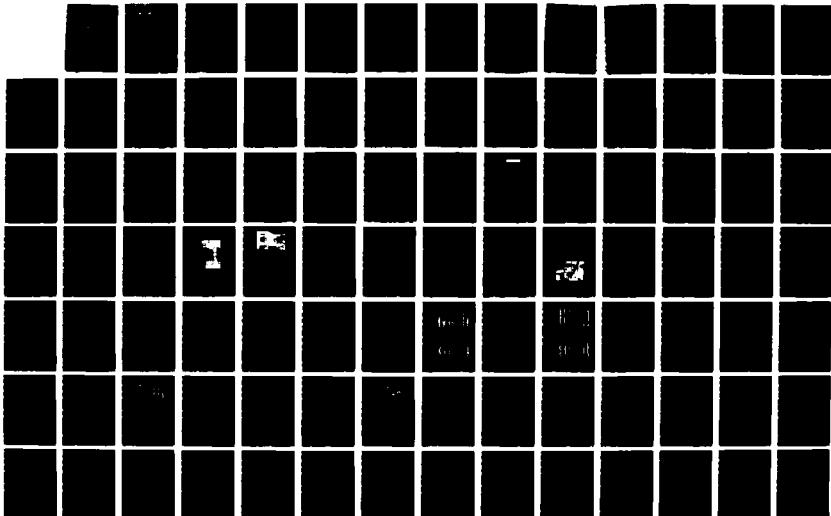
ELECTROMAGNETIC METRICS OF MENTAL WORKLOAD(U) PURDUE
UNIV LAFAYETTE IN EEG SIGNAL PROCESSING LAB
J I AUNON ET AL SEP 87 AFOSR-TR-87-1663 AFOSR-85-0313

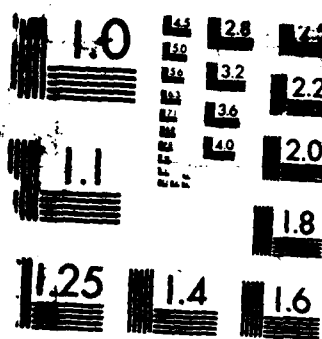
123

UNCLASSIFIED

F/G 5/8

NL





MICROCOPY RESOLUTION TEST CHART

UNCLASSIFIED
SECURITY CLASSIFICATION

AD-A188 205

ATION PAGE

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

3 DISTRIBUTION/AVAILABILITY OF REPORT

2b DECLASSIFICATION/DOWNGRADING SCHEDULE

Distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)

1285-0604

5 MONITORING ORGANIZATION REPORT NUMBER(S)

AFOSR-TR- 87-1663

6a NAME OF PERFORMING ORGANIZATION
Purdue Research Foundation
Division of Sponsored Programs

6b OFFICE SYMBOL
(if applicable)

7a NAME OF MONITORING ORGANIZATION

AFOSR/NL

6c ADDRESS (City, State, and ZIP Code)

Hovde Hall
West Lafayette, IN 47907

7b ADDRESS (City, State, and ZIP Code)

Building 410
Bolling AFB, DC 20332-6448

8a NAME OF FUNDING/SPONSORING ORGANIZATION
Air Force
Office of Scientific Research

8b OFFICE SYMBOL
(if applicable)
NL

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

AFOSR-85-0313

8c ADDRESS (City, State, and ZIP Code)

Building 410
Bolling AFB, DC 20332-6048

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
61102F	2313	A4	

11 TITLE (Include Security Classification)

Electromagnetic Metrics of Mental Workload (U)

12 PERSONAL AUTHOR(S)

Aunon, J. I.; Kantowitz, B. H.; McGillem, C. D.; Plonski, M. P.

13a TYPE OF REPORT
Final

13b TIME COVERED
FROM 8/15/85 TO 8/14/86

14 DATE OF REPORT (Year, Month, Day)
1987 September

15 PAGE COUNT
222

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Workload Heart Rate
EEG Respiration
EP SQUID

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Mental workload has been very difficult to describe quantitatively. An excessive workload can lead to a decrease in accuracy and performance, while a sustained high level of workload can lead to mental exhaustion. Previous research has indicated that heart rate variability and evoked potentials in the EEG (electroencephalogram) may be linked to mental workload. Unfortunately most of the work to date has examined these two biocybernetic variables independently rather than jointly. Recent advances now allow one to measure the magnetic fields produced by the brain (MEG) using a SQUID magnetometer (Superconducting Quantum Interference Device). Much of the MEG research to date has concentrated on lower order brain processes rather than the higher cognitive processes associated with workload.

The current research examines all these biocybernetic variables jointly in an effort to quantify mental workload. A paradigm was developed to vary several aspects of mental workload and verify the "hybrid capacity" model of human information processing that was

(continued)

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT

☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION

UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL

Dr. Alfred R. Fregly

22b TELEPHONE (Include Area Code)

202-767-5021

22c OFFICE SYMBOL

AFOSR/NL

19. ABSTRACT (continued)

developed at Purdue. This research included constructing a data acquisition system to implement this paradigm and simultaneously record heart rate, respiration, EEG & MEG data. It is believed that this is the first time that such a varied data set was recorded simultaneously. ←

Behavioral measures of response did not match the loading levels predicted by the model for all conditions. It did match though for the higher levels of loading. Power spectral estimates of the prestimulus interval and simple averaging did not appear to give useful measures of the loading. The MEG data was highly corrupted due to head movement. The use of latency corrected averaging (LCA) allowed the identification of peaks (P300 & N200) which were capable of distinguishing the loading levels that showed significant differences in the behavioral measures of performance. The heart rate variability was able to distinguish between the same loading levels that the behavioral variables and EEG peaks could.



AFOSR-TK- 87 - 1663

Electromagnetic Metrics of Mental Workload

**J. I. Aunon
B. H. Kantowitz
C. D. McGillem
M. P. Plonski**

**EEG Signal Processing Laboratory
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907**

**AFOSR-85-0313
September 1987**

**Prepared for
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
Bolling Air Force Base, DC 20332**

Final Report

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES.....	vi
ABSTRACT	ix
CHAPTER 1 - INTRODUCTION.....	1
1.1 What is Loading.....	1
1.2 Measures of Mental Load	1
1.3 Biocybernetic Variables and Experimental Procedure.....	3
1.4 Data Analysis and Results	4
CHAPTER 2 - HUMAN INFORMATION PROCESSING	5
2.1 Introduction	5
2.2 Double Stimulation	5
2.3 Limited - Capacity Channel Model.....	6
2.4 Stage Models	7
2.5 Stage Capacities.....	10
2.6 Variable Capacity Model	11
2.7 Dual Task Stimulation	11
2.8 Hybrid Capacity Model.....	14
CHAPTER 3 - BIOCYBERNETIC VARIABLES.....	17
3.1 Heart Rate	17
3.1.1 Introduction.....	17
3.1.2 Heart Rate Variability	18
3.1.3 Spectral Techniques	19
3.2 Respiration	22

3.3 Event Related Potentials	22
3.3.1 Introduction.....	22
3.3.2 Measurement Techniques for ERP	24
3.3.3 Classification of ERP.....	25
3.3.4 Cognitive Uses of ERP	27
3.4 Biomagnetism.....	28
3.4.1 Introduction.....	28
3.4.2 SQUIDS	30
3.4.3 Models for Evoked Fields	34
CHAPTER 4 - EXPERIMENTAL PROCEDURE.....	37
4.1 Introduction.....	37
4.2 Paradigm	37
4.3 Physiological Measurements.....	42
4.3.1 Overview	42
4.3.2 Respiration & Heart Rate	42
4.3.3 Electroencephalogram and EOG.....	43
4.3.4 Magnetoencephalogram.....	43
4.4 Data Acquisition System.....	45
CHAPTER 5 - PRELIMINARY DATA ANALYSIS.....	48
5.1 Introduction.....	48
5.2 Behavioral Responses	48
5.3 Heart Rate	51
5.4 Respiration	56
5.5 Artifact Rejection and Prefiltering	58
5.6 Power Spectral Estimate of the Prestimulus Interval.....	60
5.7 Average of the ERP and the EF	61
CHAPTER 6 - ADVANCED ANALYSIS	65
6.1 Introduction.....	65
6.2 Spectral Estimates of Heart Rate Variability.....	65
6.3 P300 as a Measure of Loading	83
6.4 The Latency Corrected Average	87
6.5 Conclusions.....	89
LIST OF REFERENCES.....	96

APPENDICES

Appendix A: Averages and Power Spectral Estimates.....	81
Appendix B: "MUTUAL" - SQUID Calibration Program	112
Appendix C: Data Acquisition Code.....	119



Accession For	
NTIS CANAL	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Availability for Special
A-1	

LIST OF TABLES

Table	Page
4.2.1 Subject Summary	41
4.2.2 Test Sequence for Subjects.....	41
5.2.1 Behavioral Performance Predicted by Task Description	53
5.2.2 Measured Behavioral Performance.....	53

LIST OF FIGURES

Figure	Page
2.3.1 Limited Capacity Channel.....	8
2.4.1 Stage Models	8
2.4.2. Serial & Parallel Element Processing.....	9
2.6.1. Variable Allocation Model	9
2.7.1 Primary & Secondary Task Interaction	13
2.7.2 Dual Task Interactions on Performance	13
2.7.3 Combined Dual Task Effect on Performance	15
2.8.1 Hybrid Capacity Model.....	15
3.1.1 Components of Heart Rate Variability	21
3.2.1 Sinus Arrhythmia as Detected by a Cardiotachometer	23
3.2.2 Model for Components of Sinus Arrhythmia.....	23
3.3.1 Linear Classifiers	26
3.4.1 Bioelectric and Biomagnetic Fields	29
3.4.2 Magnetic Sensing Devices.....	31
3.4.3 DC SQUID and Mounting Systems.....	32
3.4.4 Shielded Room.....	33

Figure	Page
3.4.5 Schematic for a DC SQUID System	33
3.4.6 Detector Coil Arrangements.....	35
4.2.1 Single Trial Timing Diagram.....	38
4.2.2 Test Setup with Subject.....	38
4.2.3 Experiment Sequence.....	40
4.4.1 Equipment Configuration	46
5.2.1 Response Time to 1st Stimulus (RT1).....	50
5.2.2 Response Time to 2nd Stimulus (RT2).....	50
5.2.3 Percent of Correct Responses to Both Stimuli.....	52
5.3.1 Mean Heart Inter-Beat-Interval (IBI).....	52
5.3.2 Standard Deviation of Heart IBI.....	55
5.3.3 Standard Deviation of Heart IBI (MSSD Method)	55
5.3.4 Respiratory Sinus Arrhythmia as Detected using IBI	57
5.3.5 IBI and RT1 over one Complete Block	57
5.4.1 Mean Respiratory Intervals.....	59
5.4.2 Standard Deviation Respiratory Intervals	59
5.7.1 Stimulus Aligned & Response Aligned: Average P3.....	63
5.7.2 Stimulus Aligned & Response Aligned: Average MEG.....	63
6.2.1 The Low Pass Filtered Event Series.....	68
6.2.2 PSE for Original IBI vs PSE for LPFES ($dt = \text{mean IBI}$).....	69
6.2.3 PSE for Original IBI vs PSE fo LPFES ($dt = 1 \text{ sec}$).....	69

Figure	Page
6.2.4 LPFES Spectrum for Various Prefiltering	72
6.2.5 Total Power (DC to 0.5 Hz)	75
6.2.6 Power in the Thermal Band	76
6.2.7 Power in the Blood Pressure Band	77
6.2.8 Power in the Stimulus Band	78
6.2.9 Power in the Respiration Band	79
6.2.10 Percent Power in the Bands	81
6.2.11 Relative Power in the LPFES Spectrum	82
6.3.1. 3.5 Hz LPF Averages: P3-Electrode, Single Stimulation	84
6.3.2. 3.5 Hz LPF Averages: P3-Electrode, 250msec ISI	85
6.3.3. 3.5 Hz LPF Averages: P3-Electrode, 62.5msec ISI	86
6.4.1. LCA Peaks: P3-Electrode, 62.5msec ISI	90
6.4.2. LCA Peaks: P4-Electrode, Subject 6	91
6.4.3. LCA Peaks: P4-Electrode, Subject 7	92
6.4.4. LCA Peaks: P4-Electrode, Subject 8	93
6.4.5. Response Time 1 vs P300 Latency	94
Appendix	
Figure	
B1 Predicted vs Measured Coupling	115
C1 Equipment Configuration	128
C2 Single Trial Timing Diagram	129
C3 Interrupt Arbitration Circuit	130

ABSTRACT

Mental workload has been very difficult to describe quantitatively. An excessive workload can lead to a decrease in accuracy & performance, while a sustained high level of workload can lead to mental exhaustion. Previous research has indicated that heart rate variability and evoked potentials in the EEG (electroencephalogram) may be linked to mental workload. Unfortunately most of the work to date has examined these two biocybernetic variables independently rather than jointly. Recent advances now allow one to measure the magnetic fields produced by the brain (MEG) using a SQUID magnetometer (Superconducting Quantum Interference Device). Much of the MEG research to date has concentrated on lower order brain processes rather than the higher cognitive processes associated with workload.

The current research examines all these biocybernetic variables jointly in an effort to quantify mental workload. A paradigm was developed to vary several aspects of mental workload and verify the "hybrid capacity" model of human information processing that was developed at Purdue. This research included constructing a data acquisition system to implement this paradigm and simultaneously record heart rate, respiration, EEG & MEG data. It is believed that this is the first time that such a varied data set was recorded simultaneously.

Behavioral measures of response did not match the loading levels predicted by the model for all conditions. It did match though for the higher levels of loading. Power spectral estimates of the prestimulus interval and simple averaging did not appear to give useful measures of the loading. The MEG data was highly corrupted due to head movement. The use of latency corrected averaging (LCA) allowed the identification of peaks (P300 & N200) which were capable of distinguishing the loading levels that showed significant differences in the behavioral measures of performance. The heart rate variability was able to distinguish between the same loading levels that the behavioral variables and EEG peaks could.

CHAPTER 1

INTRODUCTION

1.1. What is Loading

As engineering systems become more complex, the man-machine interface may become the limiting factor to increased performance. This is especially evident in complex real time interfaces such as between a pilot and a modern airplane. Cognitive psychologists have developed many models of human information processing (review in Kantowitz, 1985a, 1974; Taylor, 1976; Townsend, 1974; Broadbent, 1971; Sternberg, 1969). In order to justify the model, they must design experiments with human subjects geared to test hypotheses predicted by that model. The measurement of workload is of central importance to these applications.

The term "load" or "workload" has many meanings associated with it and it is necessary for us to first clarify our definition of it before we can attempt to measure it. Weiner (1982) describes workload in the context of mechanical stress and strain. As long as the body's regulatory systems are able to compensate for the load (e.g. increased cardiac output), the subject is said to be stressed. When the body is overloaded and can no longer fully compensate (e.g. build up of lactic acid), the subject is said to be under strain. This distinction becomes harder to make for psychological systems. Overload is generally perceived as a change of emotional state (e.g. anxiety). A decrease in performance accuracy has been used in the context of overload, however it has also been associated with a "high workload".

1.2. Measures of Mental Load

While cardiac output and oxygen uptake may provide a good measure of physical workload there is no corresponding measure for mental workload. This is primarily because there is no adequate model for human information processing. It is difficult to come up with a measure for something, when we

are not even sure as to what we wish to measure. Three different approaches are generally used to detect mental load (Kalsbeek, 1973; Gopher & Braune, 1984):

- 1) personal experience.
- 2) objective description of the task parameters
- 3) measures of response (either behavioral or physiological)

This is a starting point for clarifying the concept of mental workload, but there are still many shortcomings.

Personal experience is probably more a measure of perceived cognitive difficulty than actual loading. Hart & Bortolussi (1984) used a pilot opinion survey to create a database of flight scenarios with predetermined levels of pilot workload. Pilots were asked to rate their workload and other parameters as they went through the predetermined flight scenarios. There was a high degree of correlation between the perceived workload, stress and effort ratings as one would expect in such a circular approach to measuring workload. What is of interest to note though, is that the performance ratings were not correlated with the other measures. Gopher & Braune (1984) report similar findings in a different experiment and conclude with *"It was hinted that subjective measures may be more related to the formal properties of the task than to the details of performance."*

Kalsbeek (1973) also discusses *"whether the concept of mental load is indivisible."* It may be that a single measure of loading is not sufficient to describe the total cognitive processing involved with a given task. Some argue (Moray et al., 1979, p.105) that the only true measure of workload is the subject's own interpretation of his loading, but this has many limitations. Subjects do not have difficulty assigning a number to a given task, but how does one select the range of numbers to choose from. Is the range to be fixed and invariant with time? A student may rate the first exam as a 10 on a difficulty scale of 1 to 10. When the second and more difficult exam is given, is it then ranked as a 15 on the same 1 to 10 scale or do we rescale all previous interpretations of loading so that we do not exceed our allowable range? How do we compare ratings between students who may base their evaluations on completely different aspects of the same task. One student may base his rating on the time aspects of the test, while another may base his on the style of the exam, while yet another may base his on the actual questions asked.

This problem of multiple workload aspects for a given task is not unique to the subjective method of load description. The load can also be described in terms of the task parameters, but how does one compare the workload of responding to a pair of four choice stimuli given 250 msec apart with a pair of two choice stimuli given 62.5 msec apart. Krol & Opmeer (1971) report different trends in cardiovascular activity for cockpit workload during simulated landings and the mental load of parachute jumpers. They differentiate this as informational and emotional load. A single measure of load appears inadequate to describe the entire spectrum of cognitive processing, so how then do we proceed in our quest to develop metrics for workload classification.

One possible solution is to develop a simplified model of cognitive processing and define our workload metric in light of this model. Chapter 2 reviews some of the earlier information processing models leading to the development of the "*Hybrid Capacity Model*" (Kantowitz & Knight, 1976) which is the model used for the current research.

1.3. Biocybernetic Variables and Experimental Procedure

The current research utilizes the heart inter-beat-interval (R to R intervals of the electrocardiogram), the respiratory rate, event related potentials (electroencephalograms) and evoked fields (magnetoencephalograms) to gain an understanding of mental workload. Chapter 3 reviews the measurement of these biocybernetic variables and their application to workload studies.

In Chapter 4 the experimental setup and paradigm are discussed. A double stimulation task which also varied the number of choices was used for the current study. This allows us to vary the workload by increasing the rate that information is presented as well as varying the degree of uncertainty in the stimulus. Response times and accuracy are recorded throughout the experiment so that we have converging measures (Kantowitz, 1985b); i.e. workload is described by both task description and behavioral response.

An extremely powerful, yet flexible data acquisition system was designed and constructed to carry out this research. The system is discussed in Chapter 4 and in Appendix B. The experiment was performed in the summer of 1986 at the Aerospace Medical Research Laboratory (AMRL) at Wright Patterson Air Force Base (WPAFB).

1.4. Data Analysis and Results

The data analysis is divided into two chapters in this report. Chapter 5, entitled "Preliminary Data Analysis" and Chapter 6, entitled "Advanced Analysis". The results and comparisons to previous work are discussed in Chapter 7.

The preliminary analysis comprised of prescreening the data to reject eyeblink contamination, amplifier saturation, ectopic heart beats, etc. Power spectra were computed from the prestimulus interval for the event related potentials (ERP) and the evoked fields (EF). Average responses were computed. Mean heart rates and variances were computed.

The advanced analysis included using latency corrected averaging (Aunon & McGillem, 1977, 1978) on the ERP to separate out significant peaks in the response. Spectral techniques were used on the respiration and heart rate. Linear regressions were performed with respect to the response time.

CHAPTER 2

HUMAN INFORMATION PROCESSING

2.1. Introduction

The scope of human information processing is so broad that no single document could do it justice. The goal of this research is to develop metrics for workload and so our discussion of human information processing will be focused in this direction. Several different techniques are available for loading the human processing capabilities. The first we shall refer to as single stimulation; as the name implies, a single stimulation is presented to a subject. Loading can be varied by varying the complexity of the stimulus, i.e. locating a target element from a large versus a small set. Stimulus presentation can also be degraded to increase the loading. This technique predominantly varies the perceptual and coding aspect of information processing and is consequently not often used when one is more concerned with the total loading in an information channel.

2.2. Double Stimulation

The second method of loading is referred to as double stimulation (Kantowitz, 1974) and is the more dominant method since it allows greater flexibility in focusing the loading. Loading is accomplished by forcing the information channel to do two things at once. This can be accomplished by decreasing the interstimulus interval (ISI) between two successive stimuli to the point where the the channel does not have sufficient time to return to its unloaded state between stimulations. The workload can then be increased by decreasing the ISI and thereby increasing the amount of time that the processing will be dealing with both stimulations as opposed to a single stimulation. If the ISI becomes extremely short the two stimuli are then "grouped" and perceived as a single stimulation with two elements. This technique of loading utilizes a serial approach to double stimulation. A parallel approach can also be used by requiring the channel to handle dual tasks (primary and secondary) at the same time. Response time (RT) and

accuracy are recorded as a measure of the channel's performance. The theory states that channel performance will decrease as channel loading approaches and exceeds channel capacity. Limitations of these measures with regard to channel capacity will be discussed later. Note that depending on the task selected (e.g. continuous reading), either or both of these measures may not be available.

Cognitive psychologists construct behavioral experiments to study human information processing. Inferences are drawn from the results of these experiments about the underlying process and models are then constructed to fit these inferences. Models constructed from loading due to successive stimuli will be discussed first. These models will then be reevaluated in light of some recent results from dual task experiments.

There are several basic paradigms that can be used in a serial double stimulation (Kantowitz, 1974, p. 86). The most common of these is the *stimulus 1 - response 1, stimulus 2 - response 2* (S1 - R1, S2 - R2) paradigm since it requires a response to each stimuli. It is well known that there is delay in responding to the second stimulus and that this delay increases as the ISI is decreased. This delay has been called the psychological refractory period (PRP) since it was first believed that the human information processing channel became refractory or disabled for a brief period, similar to refractory period for nerve fibers (Telford, 1931). It is now believed that this effect has little in common with its name, but nonetheless, the effect has retained its descriptive name. This paradigm is also referred to as the PRP paradigm. Other paradigms S1, S2 - R2 and S1 - R1, S2 which do not require responses to both stimuli have also been used to verify models.

2.3. Limited - Capacity Channel Model

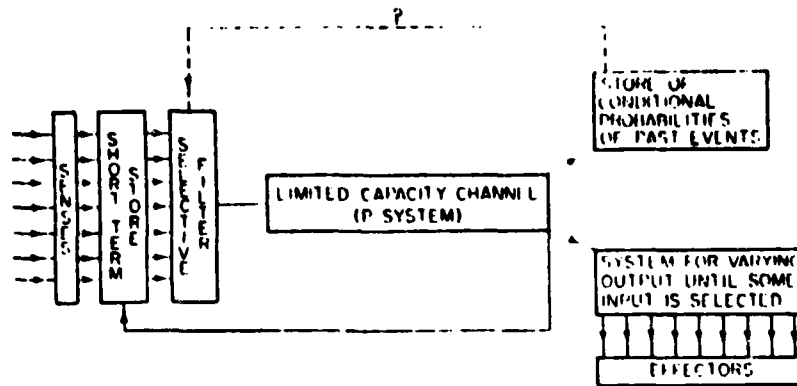
The traditional model for double stimulation is the limited capacity channel model (Broadbent, 1958; 1971) shown in Figure 2.3.1. The limited channel model was able to explain changes in reaction time due to PRP, stimulus modality and different stimuli probabilities. This became the benchmark for other models. PRP could be explained by S2 being held in short term storage while the channel was dealing with S1, since the limited capacity channel was capable of handling only one task at a time. Broadbent (1971 p. 313) now believes that delays in category selection is a better explanation. Stimuli with higher probabilities of occurrence would have faster reaction times (RT) due to the selective filter. Delays for ISI longer than the response time for stimulus 1 (RT1) could be explained since the channel

was updating memory after it had finished its response to S1. Unfortunately many earlier researchers focused on PRP, and did not consider the effects of S2 - R2 on RT1 (response time 1). It was found that RT1 increased slightly due to S2 - R2 (Herman & Kantowitz, 1970) and the limited channel capacity can not offer any explanation for this effect. Broadbent's review of the limited channel capacity model in 1971 does not explain this issue. The limited capacity channel also has difficulties explaining some effects of dual task loading which will be discussed later.

2.4. Stage Models

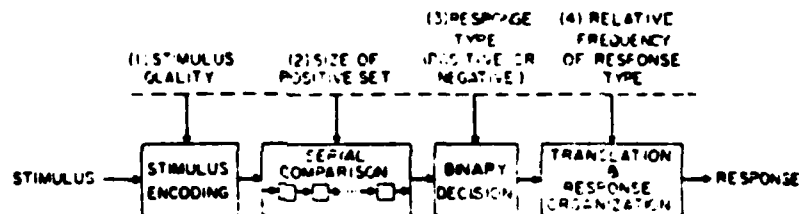
The stage approach to model human processing (Sternberg, 1969; Taylor, 1976; Townsend, 1974) has become a popular method. Figure 2.4.1 shows a simple four stage serial model of the reaction process for a binary decision. Models can be classified as serial, parallel or hybrid depending on how the stages are connected together. Kantowitz (1985b) gives an abbreviated definition of a stage as *a single transformation of information*, while Townsend (1974) gives a more precise mathematical definition.

Taylor (1976) breaks down stages into smaller terms called elements. *An element is identified with the smallest unit of information processed in a particular stage of a particular model.* The processing within each stage can then be evaluated along several lines including: parallel versus serial; exhaustive versus self terminating; and limited versus unlimited. Parallel and serial refer to whether a stage can process more than a single element at a time. Figure 2.4.2 gives an example of the time considerations for a single four element stage. Exhaustive and self terminating refers to the point at which the stage will terminate its processing. An exhaustive stage will always use the same amount of time since it will process all elements while a self terminating process will end after it processes the "critical element". A self terminating process is a possible explanation to reaction time increasing linearly with the number of choice alternatives. It will take longer on average to locate the critical element if there are more elements to search through. Unlimited refers to whether the stage can work with many elements at the same speed at which it can handle a single element. Note that parallel processing does not necessarily imply unlimited processing since the processing time for each element may increase as the number of elements increase and the resources of the stage are stretched.



A diagram of the flow of information within the nervous system, as conceived by

Figure 2.3.1 Limited Capacity Channel (Broadbent 1958)



Processing stages in binary classification. Above the broken line are shown the four factors examined. Below the line is shown the analysis of RT inferred from additive relations between factor pairs 1&2, 1&3, 2&3, 2&4, and 3&4, the linear effect of factor 2, and other considerations described in the text. The quality of the test stimulus influences the duration of an encoding stage in which a stimulus representation is formed. This representation is then used in a serial-comparison stage, whose duration depends linearly on size of positive set; in each of its substages the representation is compared to a memory representation of one member of the set. In the third stage a binary decision is made that depends on whether a match has occurred during the serial-comparison stage that precedes it; its mean duration is greater for negative than for positive decisions. The selection of a response based on the decision is accomplished in the final stage whose duration depends on the relative frequency with which a response of that type is required.

Figure 2.4.1 Stage Models (Sternberg 1969)

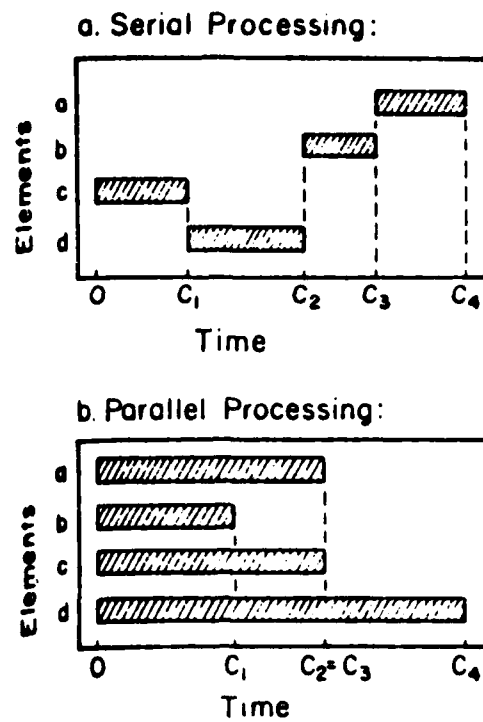
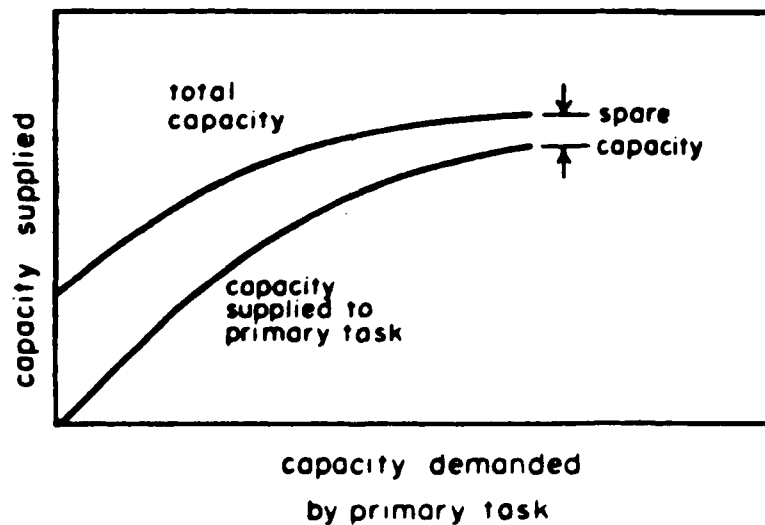


Figure 2.4.2 Serial & Parallel Element Processing (Taylor 1976)



Basic mechanism of Kahneman's variable-capacity model. Note that other variable-capacity configurations are possible, e.g., capacity supplied to primary task could continue to increase with a slope of one until the total capacity line is reached.

Figure 2.6.1 Variable Allocation Model (Kahneman 1973)

2.5. Stage Capacities

Given the complexity that even a single stage can contain, most information processing models consist of a serial channel of stages to prevent the model from getting unwieldy. The concept of the loading of an information channel is probably best described in terms of the capacity of the channel and is developed by Broadbent (1971) in his limited-capacity channel model of performance. Loading can be described as to how much of the available capacity is used. Overloading takes place when the available capacity is exceeded. Of central concern in an information channel is how the available capacity is distributed throughout the channel. Reaction time for a process is often used as an indicator of channel capacity, though this measure has many shortcomings (Broadbent, 1971; Townsend, 1974). Kantowitz (1985a) stresses caution when using RT and states: *"Ideally, lag should not be used as an index of capacity without strong converging operations."* A converging criterion would be changes in a behavioral variable which correlated with changes in physiological variables. If we accept for the moment that reaction time is a measure of capacity, then capacity allocation can be deduced by decomposing the reaction time for individual stages (Taylor, 1976).

The simplest method of decomposing reaction time is by using two tasks that differ only by a postulated stage. The difference in reaction times then represents the time for that stage. This method is known as the method of subtraction, and has fallen into disfavor though it has never been invalidated by modern experimental methods (Taylor, 1976). Sternberg (1969) proposes a method, the method of additive factors, which uses a factorial experiment to decompose reaction time. This method is very popular since it allows one to determine what factors affect what stages. The rationale of the method is that factors which interact in a factorial experiment must influence the same stage, while those which do not interact probably influence different stages. This method was used to justify the binary response model shown in Figure 2.4.1. Unfortunately the very power of this method, the ability to assign different factors to different stages, will lead to its downfall when we discuss the dual task results. The basic limitation of these methods is that they assume each stage is independent and has its own invariant source of capacity. Taylor (1976) expands upon Sternberg's method to avoid some of these assumptions, but at the cost of adding complexity to the stage description.

Kantowitz (1985a) advocates that the concept of capacity is the central topic in information processing. *"Capacity is the more important concept, since while capacity can exist without precise definition of a channel, a channel, at least in psychological research, can be inferred only from a measurement of capacity."* This heuristic solution of replacing an information channel with the notion of capacity sidesteps all the difficulties inherent in trying to define a channel. The concept of a channel will be retained in this discussion; however the term will not necessarily imply a formal description of the channel. If we are only concerned with the measurement of loading and we accept that capacity utilization is an acceptable measure of loading, then it is not necessary to further refine our model.

2.6. Variable Capacity Model

Kahneman (1973) also agrees that capacity is the central issue and has proposed a variable allocation model. The strength of this model is that it is based on a few assumptions and does not attempt to provide a structure for the processing. It only provides an abstract model to justify the effects of loading. It is very hard to disprove something that is very unspecific and makes few predictions. The basic mechanism of the variable capacity model is that total capacity increases as a function of demand. This increase is at a slower rate than the increase in capacity flowing to the primary task. This effect is illustrated in Figure 2.6.1. The channel is initially able to supply the primary (or single task) with all the capacity that it demands so that there is no decrease in channel performance for low loading. As the loading increases the channel can no longer keep up with demand and performance decreases. For dual task stimulations the amount of spare capacity for the secondary task is reduced as primary task difficulty increases, so that both task performances should decrease. Capacity is not intended to be unlimited in this model. The undefined capacity limit though is only reached in the asymptotic sense. One of the shortcomings of this model is that it does not allow any way of distinguishing between task demands and the performer's intentions (Kantowitz, 1974, p.94); i.e. how does the model reflect the relative importance of each task to the performer?

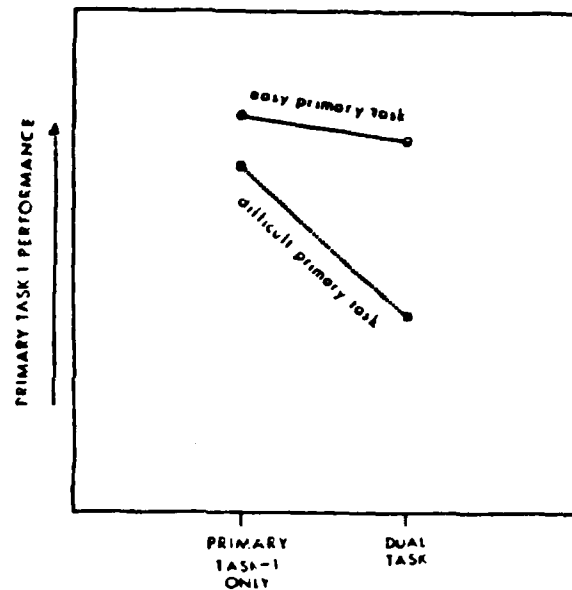
2.7. Dual Task Stimulation

Dual task stimulation (timesharing paradigm) involves the performance of two different task simultaneously. Limited capacity channels, stage models and variable capacity models all predict that performance should

decrease when the second task is added to the paradigm. This relationship is presented in Figure 2.7.1. If the addition of the secondary task causes overloading, as with the addition of the secondary task to the difficult primary task, then primary task performance will decrease considerably. If the channel does not become overloaded, as with the addition of the secondary task to the easy primary task, then there will only be a slight decrease in performance. If the two lines in Figure 2.7.1 are parallel, it implies that the same decrease in performance, due to the introduction of the secondary task, was simply added to both curves. This effect is called the additivity effect (Kantowitz, 1976). Simple additivity can be justified by all of the models discussed by either assuming that overload was not reached or that overload caused a decrease in secondary task performance.

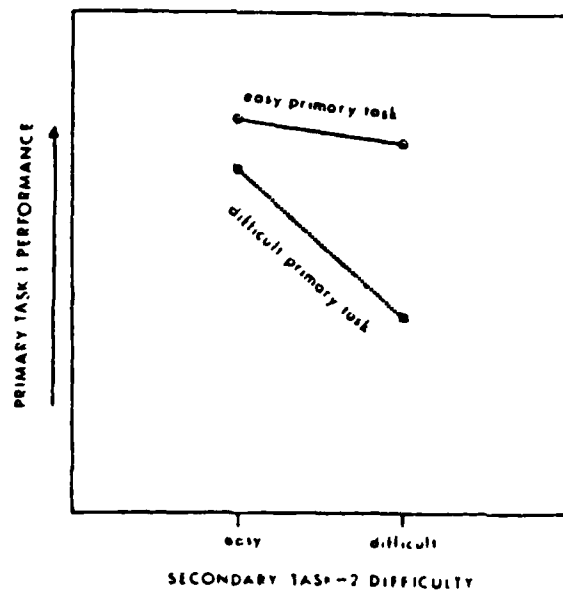
In order to fully understand timesharing paradigms, it is necessary to perform each component task at two levels and each component task alone to generate a single stimulation baseline (Kantowitz, 1974). These results can then be presented as shown in Figure 2.7.2. Note that while this graph has the same shape as Figure 2.7.1 it represents dual task performance, while Figure 2.7.2 represents the interaction of single task and dual task performance. Limited channel capacity and variable capacity models only allow for the possibility of additivity as shown in Figure 2.7.2 if one assumes that two simultaneous difficult tasks do not strain capacity. Stage analysis (Sternberg, 1969) does allow for additivity at this level by then placing each task in a different parallel stage with its own independent source of capacity.

The results of an experiment performed by Kantowitz & Knight (1974, 1976) demonstrates the use of these two graphs. They obtained additivity for dual task interactions (as in Figure 2.7.2), but overloading for single task versus dual task (as in Figure 2.7.1). If one takes liberties (i.e. considering no dual task to be a very easy task) with these graphs, the results of Kantowitz & Knight's experiment can be summarized in a single graph as in Figure 2.7.3. The inconsistencies with the different models are apparent in this graph. The first part of this graph implies that capacity is strained in going from the single task to the dual task, but that capacity is no longer strained as we increase the difficulty of the dual task. The limited capacity channel and the variable capacity channel are not able to explain this result. The stage approach also fails since the first part of the graph predicts (via the method of additive factors) that the primary and secondary task are located in the same stage, while the second part of the graph predicts that they are



Predictions of limited-capacity model when single and dual-task conditions are compared. There are two levels (Easy and Hard) of the primary task (also called task-1). Task-1 only represents primary task performance when no secondary task is required. Dual-task represents primary task performance when one level of a secondary task is simultaneously required. Performance on the secondary task is not shown in the figure.

Figure 2.7.1 Primary & Secondary Task Interactions (Kantowitz&Knight 1976)



Predictions of limited-capacity model for dual-task performance. Two levels of the primary task are combined with at least two levels of a secondary task.

Figure 2.7.2 Dual Task Interactions on Performance (Kantowitz&Knight 1976)

located in different stages. While the existing models could be modified to incorporate these results, an alternate approach would be to construct a new model based on these findings.

2.8. Hybrid Capacity Model

A new model (Figure 2.8.1), is introduced by Kantowitz & Knight (1976) to explain these findings. Although the authors refer to it simply as a hybrid model, we shall refer to it as the hybrid capacity model. This model is basically a conglomeration of the other models. The static capacity allocator (SCA) can be used to divide capacity in accordance with the performers intentions (e.g. relative task importance). The entire channel has a single limited source of capacity which is dynamically allocated to some stages directly and to some stages through the SCA. This implies that in stages 1 and 2 of Figure 2.8.1 the ratio of capacities between these stages is set at the start of the experiment, but that the actual levels of capacity will change dynamically with task demands.

The model behaves as a serial stage for low workloads where the total capacity required does not exceed the available supply. It is not necessary for the limited capacity source to dynamically trade off capacity, so that each stage appears to have its own independent supply of capacity. As workload is increased, capacity available to each stage is not sufficient to keep up with demand and the model behaves as a limited capacity channel.

This model appears to place the bottleneck of the processing channel at the later output stage since this is where several parallel stages feed a single stage. There are many different views as to where capacity limitations cause bottlenecks (Kantowitz, 1974), and some even argue that there are no bottlenecks (Kahneman, 1973). This arrangement was probably a personal bias of the author as expressed by Kantowitz (1985a p. 160): *"Indeed, the present author is guilty of this oversimplification by claiming that response processes are a more important locus of limitation than are stimulus processes. While this may be true (e.g. I still believe it), it ignores implications of capacity limitations from elements for limitations upon stages."* Note that this statement was not made in reference to any model in particular, but rather during a discussion of capacity limitations in stages.

This model is applicable to both PRP and timesharing paradigms. The parallel stages controlled by the SCA are not necessarily a primary and secondary task as in timesharing paradigm. They can also be used to

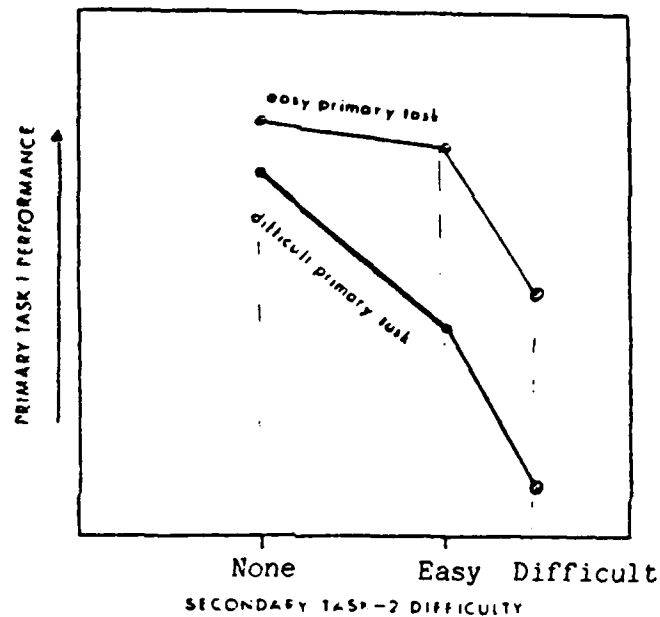
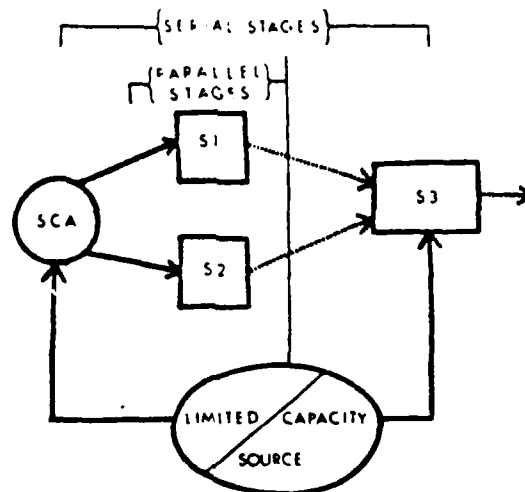


Figure 2.7.3 Combined Dual Task Effect on Performance



Schematic rendering of a hybrid model. Solid lines represent capacity allocation to stages while dotted lines represent information flow between stages. The information flow inputs to stages 1 and 2 have been deleted to improve the clarity of the diagram. A limited capacity source dynamically feeds both stage S3 and a static capacity allocator (SCA) which partitions capacity between stages S1 and S2. Stages S1 and S2 operate in parallel and stage S3 operates in serial with both preceding stages. Each stage can potentially be broken down into smaller stages depending upon the level of analysis. For example, stage S3 is a molar representation of response selection, execution and control processes.

Figure 2.8.1 Hybrid Capacity Model (Kantowitz & Knight 1976)

represent attention and arousal effects. It is well known that accuracy can be traded off for speed (Pachella, 1974). The SCA could probably be used with some postulated states to explain this effect. It may also be possible to explain effects of stimulus quality on RT by using the SCA.

While this model does not give any insight as to how information is actually processed in a channel, we must remember that the focus of this research is on workload (i.e. capacity) measurement and this model will be sufficient for our needs. We shall simply refer to the stages in this model as early and late stages of processing without worrying specifically about what processing is performed in them.

CHAPTER 3

BIOCYBERNETIC VARIABLES

3.1. Heart Rate

3.1.1. Introduction

Heart rate (HR) is normally variable over time in a relaxed individual. HR is easily determined by computing the R-R interval from an electrocardiogram (ECG). A lead II configuration (Guyton, 1981) is used in order to enhance the R wave for easy computer identification. One electrode is placed on the upper right chest and the other on the left ankle with the right ankle used as a ground. The signal is amplified and then threshold detected by the computer to determine the exact time of occurrence for each R wave.

Numerous regulatory mechanisms are constantly competing amongst themselves to maintain the circulatory system in what can best be described as a state of dynamic equilibrium. Heart rate, blood pressure and cardiac output are intimately related and must be able to respond to varying demands that the body may place on them. The effect of these regulatory mechanisms on the heart rate has been an area of significant research in the study of mental workload. The heart rate is slowed through parasympathetic (vagal) stimulation of the heart which primarily slows the cardiac pacemaker (sino-atrial node). Sympathetic mechanisms are more far reaching and affect the systemic system as well as the heart. Note that since the heart is regulated by both sympathetic and parasympathetic mechanisms, it is possible for identical HRs to arise from a variety of circumstances. The specifics of these mechanisms can be found in any good physiology book such as Guyton (1981).

3.1.2. Heart Rate Variability

There has been some confusion between psychologist and physiologist over the term sinus arrhythmia (SA). For the purposes of this report, the term SA shall be used to refer to the variation of heart rate due to respiratory effects and the term heart rate variability (HRV) to refer to any variation from a constant heart rate. The term HRV shall not refer to any specific method of numerically computing the variation in heart rate since numerous methods (Firth, 1973) have been used to emphasize different aspects of the variability. Interest in using HRV as a measure of mental load arose after Kalsbeek & Ettema (1963) reported that HRV was "*gradually suppressed when increasing the difficulty of the task.*" Since that time there have been numerous articles (Kalsbeek, 1973; Porges et al., 1980; Akselrod et al., 1981; Sharit & Salvendy, 1982) and a symposium leading to an entire issue of *Ergonomics* (Rolfe, 1973) devoted to the issue of HR variability. Unfortunately this proliferation of research has led to much confusion, since the problem is not as straight forward as it might have appeared after Kalsbeek's initial findings.

Many researchers simply started measuring HR under real life working conditions and tried to draw premature conclusions about workload. While changes were found in mean heart rate (MHR), instantaneous heart rate and variance, there are numerous psychological factors which could account for these changes under normal work environments (Firth, 1973). Some researchers (Sharit & Salvendy, 1982; Heslegrave et al., 1979) advocate the use of the mean square successive differences (MSSD) as a measure of HRV and the MHR instead of the sample variance of the interbeat interval. Mulder (1973) used numerous measures of HR variability: MHR, second, third & fourth moments of R-R interval, sum of absolute differences of R-R interval, number of reversal points, etc. in an attempt to measure workload. It was found that there were many side effects of the task (e.g., attention motility, motor load from button pushing, respiration, stimulus frequency) which contributed to overall changes in HR variability. This is one of the major stumbling blocks to understanding how mental load affects HR.

Mulder (1973) concludes his work with "*In future research more attention must be paid to the types of tasks and the behavioral mechanism involved, to their presentation mode and to a more sophisticated analysis of heart rate variability.*" Sharit & Salvendy (1982) used two contrasting tasks, one was largely based on visual perception (low mental load) and the other required the mental solution (high mental load) of arithmetic problems.

MHR appeared to be far more sensitive than HRV in distinguishing between these two tasks. This is in contrast to Kalsbeek & Etterma (1963) who contend that HRV should diminish with increased load, but that MHR should remain relatively constant. It was contended that the differences were due to attentional loads versus informational loads. These results suffer from the lack of a converging criterion (Kantowitz, 1985a, 1985b) as discussed elsewhere in this report.

The lack of a precise definition of mental workload still clouds much of the issue. The only way to verify that HRV depends on the workload is to vary the workload, but that is exactly what we do not know how to measure. A behavioral measure is needed to provide the converging criterion. The level of vigilance and arousal are generally considered to be related to mental load. It has also been shown (Kalsbeek, 1963) that noise and open eyes tended to suppress some of the HRV found with the eyes closed. It may be that HRV is responding more to arousal levels than workload levels. While the autonomic controls which regulate the cardiovascular system are well understood, it is not clear how workload affects these regulatory mechanisms.

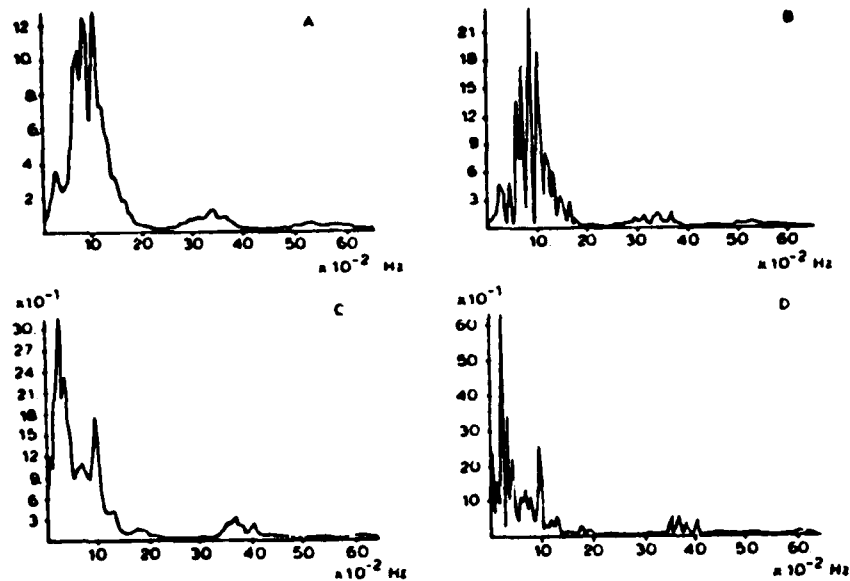
3.1.3. Spectral Techniques

The most promising area of research in HR variability appears to lie with spectral processing techniques (Sayers, 1973, 1975; Mulder et al., 1973; Porges et al., 1980; Akselrod et al., 1981; DeBoer et al., 1984; Sharit & Salvendy, 1982; Kantowitz, 1985b). The R-R interval is the variable that is usually recorded in the course of an experiment. The data can be interpolated to form an equally time spaced data set and the power spectrum estimate (PSE) of the interbeat interval is then computed (Kantowitz, 1985b; Mulder et al., 1973). Others (DeBoer, 1984; Sayers, 1975) compute the PSE directly from the R-R intervals, by considering the series to be equally beat sampled (rather than equally time sampled), or by direct transform techniques, which do not rely on samples being equally spaced. DeBoer (1984) has shown that these two direct methods are intimately related and should lead to equivalent results when all effects are accounted for. The suppression of HRV activity shows up as a reduced PSE with the remaining energy shifting closer to the lower frequencies. Spectral techniques allow us to use several measures of HRV (Kantowitz, 1985b) from different bands of the PSE which are associated with different regulatory mechanisms.

Figure 3.1.1 shows a PSE of the interbeat interval computed from workers during a period of low activity and from a period of industrial workload of an unreported nature. The components in Figure 3.1.1 related to respiratory activity (≈ 0.35 Hz), vasomotor (blood pressure) activity (≈ 0.1 Hz) and thermal activity (≈ 0.025) were determined by independently correlating these variables previous to collecting this data. The respiratory activity affects the blood pressure as well as the heart rate. The activity at 0.1 Hz related to vasomotor activity refers to the internal oscillatory nature of the blood pressure regulatory system. The blood pressure regulatory system is complex and in part depends on the property of smooth muscle tissue and neural control from the brain stem. Due to the nature of the regulatory mechanism, it is possible for an external periodic signal to entrain the oscillation (replace the normal oscillation with the external driving frequency). This effect is similar to that of the hierarchy of cardiac pacing or that of a synchronizable electronic oscillator. Respiration has been known to entrain the blood pressure regulation under certain conditions (Sayers, 1973). Since the brain stem is an integral part of the blood pressure regulatory mechanism, it may be that mental loading may somehow influence this regulation. Figure 3.1.1 shows that vasomotor activity is greatly reduced with the increased loading, but one should not jump to premature conclusions.

Spectral techniques not only allow us to identify some of the external influences on HR irregularity, they also allow us to correct for these effects (Bendat & Piersol, 1971 chap. 5; Porges et al., 1981). Caution should be exercised before we blindly correct for all these effects. The mechanisms of mental workload are not clearly understood and the primary effect of loading may indeed be on the regulatory mechanism. The regulatory mechanisms described above account for 82% (Sayers, 1973) of the variance in the HR. Clearly there are some effects that should be removed. Mulder et al. (1973) reports a peak in the power spectrum at the stimulus frequency. This effect should be removed since it is a function of the task and not the loading. The effects of respiration are often removed before computing the variance (Porges et al., 1981; Kantowitz, 1985). Any effect of workload on respiration and its effect on autonomic control is then lost. If one deals with the PSE rather than merely the variance it is not necessary to categorically remove the effects of the autonomic control, since they can be examined on an individual basis.

Clearly there are workload effects on the HR variability, but more study needs to be done to isolate these effects. Experiments should be designed to



Spectra of two successive sets of 256 interbeat intervals in smoothed form (A,C) unsmoothed (B,D), with frequencies expressed in Hz. Respiratory activity (near 0.35 Hz), vasomotor activity (near 0.1 Hz) and thermal activity (near 0.025 Hz) can be identified. Curve A relates to a period of little activity whereas curve C relates to the subsequent 3-min period, at the start of an industrial workload situation.

Note that the ordinates of each curve are normalised individually to set the largest component to a specific height. The thermal components are substantially steady but the 0.35 Hz band respiratory activity diminishes substantially from A to C; similarly there is a major reduction in presumed vasomotor components in the 0.07-0.10 Hz region. The normalisation procedure clarifies the relative spectral shift towards lower frequency components.

Figure 3.1.1 Components of Heart Rate Variability (Sayers 1973)

minimize the effects of parameters unrelated to workload. The use of converging operations will also lend more weight to any results obtained. It may prove that HR variability has limited usefulness in real world workload situations until we have a better understanding of the underlying effects.

3.2. Respiration

The respiratory rate is usually measured in workload studies so that the effect of sinus arrhythmia (SA) on the HR can be removed or accounted for. During normal respiration, the heart rate may swing by 5% due to the respiratory cycle, but this may increase to 30% during deep breathing as shown in Figure 3.2.1 (Guyton, 1981). Respiratory rate is normally quite variable so that its use as a measure of mental workload may lie more in its relation to the HRV than as an independent measure by itself.

Since the central nervous system (CNS) is an integral part of the autonomic regulation, loading of the CNS may have some effect on the performance of the autonomic regulation. Inspiration causes an occlusion of the veins which leads to a temporary decrease in blood pressure. The carotid sinus reflex then causes the heart rate to increase in response to the occlusion of the veins. Inspiration also stimulates stretch sensors that inhibit vagal effects (Porges et al., 1981) which also causes the heart rate to increase. Respiration is also inhibited during inspiration by these stretch receptors due to the Hering-Breuer reflex. Valentinuzzi and Geddes (1974) present a model for the components of SA (Figure 3.2.2). They claim that the dominant mechanism of SA during normal conditions is the central component due to a coupling (K) between the respiratory center and cardiac center in the medulla. This conclusion is based on sustained heart rate variability during voluntary apnea (holding breath) in humans and drug induced apnea in dogs. Another possibility is that of independent variations of frequency within the cardiac center itself. If these mechanisms are affected by workload, then the loading may show up as a change in the heart rate itself or in the relationship between heart rate and respiration.

3.3. Event Related Potentials

3.3.1. Introduction

There has been such a cornucopia of research in the area of signal processing of electroencephalogram (EEG) and event related potentials (ERP) that it is difficult to establish a starting point for this discussion. The most



Sinus arrhythmia as detected by a cardiotachometer. To the left is the recording taken when the subject was breathing normally; to the right, when breathing deeply.

Figure 3.2.1 Sinus Arrhythmia as Detected by a Cardiotachometer
(Guyton 1981)

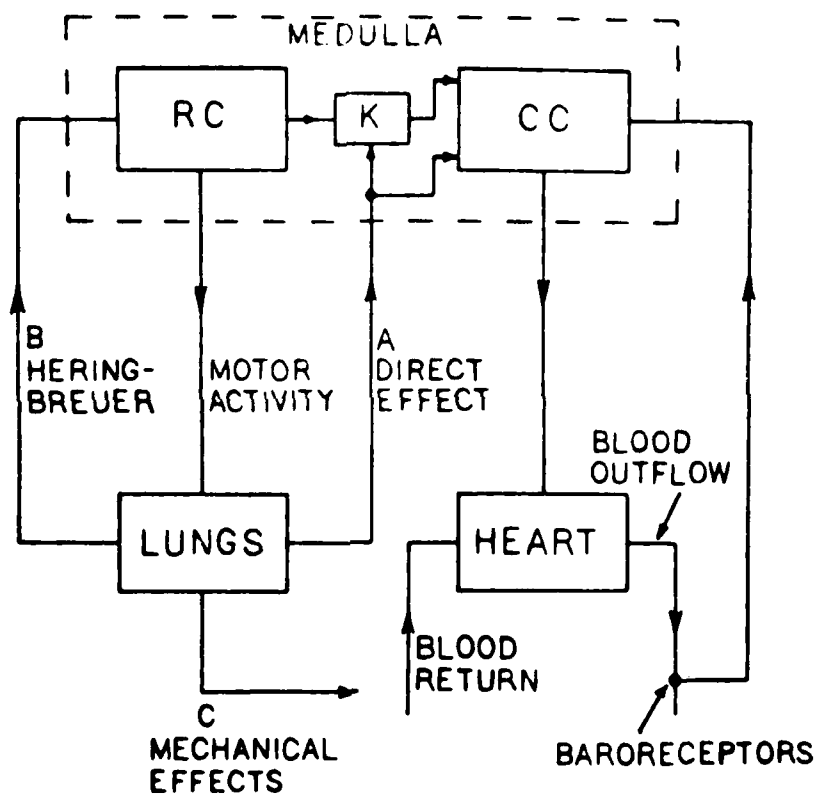


Figure 1. Components of the respiratory heart-rate response. 1) Central components, periodic influence from respiratory centers (RC) via coupling (K) to cardiac centers (CC), or oscillations in CC, or both. 2) Reflex components: (A) direct effect on CC, or on K, or both, from the stretch receptors of the lungs; (B) effect on RC, transmitted to CC through K and originating also at the pulmonary stretch receptors (i.e., an extension of the Hering-Breuer reflex); (C) mechanical effects via changes in arterial pressure and blood return to the heart.

Figure 3.2.2 Model for Components of Sinus Arrhythmia
(Valentinuzzi & Geddes 1974)

recent and broadest review on the subject of "electromagnetic signal of the human brain" was done by Gevins (1984), though a more specific three part review on "signal processing in evoked potential research" was done in 1981 (Aunon et al.; Childers et al.; McGillem et al.). In order to limit this section to a reasonable size, we will limit our discussion to methods that are relevant to the current research only. This will primarily include discussion of the ERP with respect to:

- 1) advances in measurement technique
- 2) classification
- 3) cognitive uses

3.3.2. Measurement Techniques for ERP

It was in 1875 that Dr. Richard Canton first recorded electrical activity from the exposed surface of rabbits' and monkeys' brains (Braizer, 1984). Fifty four years later, Hans Berger recorded the first human EEG and hoped that it would provide a "window on the mind" (Gevins, 1984). One might argue that since that time we have found the window, but that it is so dirty that we can't quite make out what it is that we see.

ERPs measured from the scalp typically vary from a few tenths of a microvolt to several microvolts and are imbedded in the ongoing EEG waveform whose amplitude is typically 10 to 30 microvolts (McGillem et al., 1981). The ERP is usually considered to be a deterministic signal added to a random noise process (on going EEG) with signal-to-noise ratios (SNR) as low as -20 dB. Improvements in signal estimation have progressed in two directions, averaging techniques and filtering of single trials. Most of these techniques assume that the EEG is an independent noise process, which may not be entirely valid.

The earliest "averaging" technique was simple photographic superposition (Dawson, 1947). The most common and simplest average is the conventional straight average, which predicts an increase in the SNR by the root of the number of trials (Aunon et al., 1981a). Woody (1967) introduced the cross-correlation average which shifted each trial by the amount that would cause the cross correlation between that trial and the conventional average would have a maximum at zero lag. The shifted trials were then averaged. This allowed one to compensate for random latency shifts between trials. Aunon & McGillem (1977, 1978) expanded upon this idea and developed the latency corrected average (LCA) which allowed one to compensate for

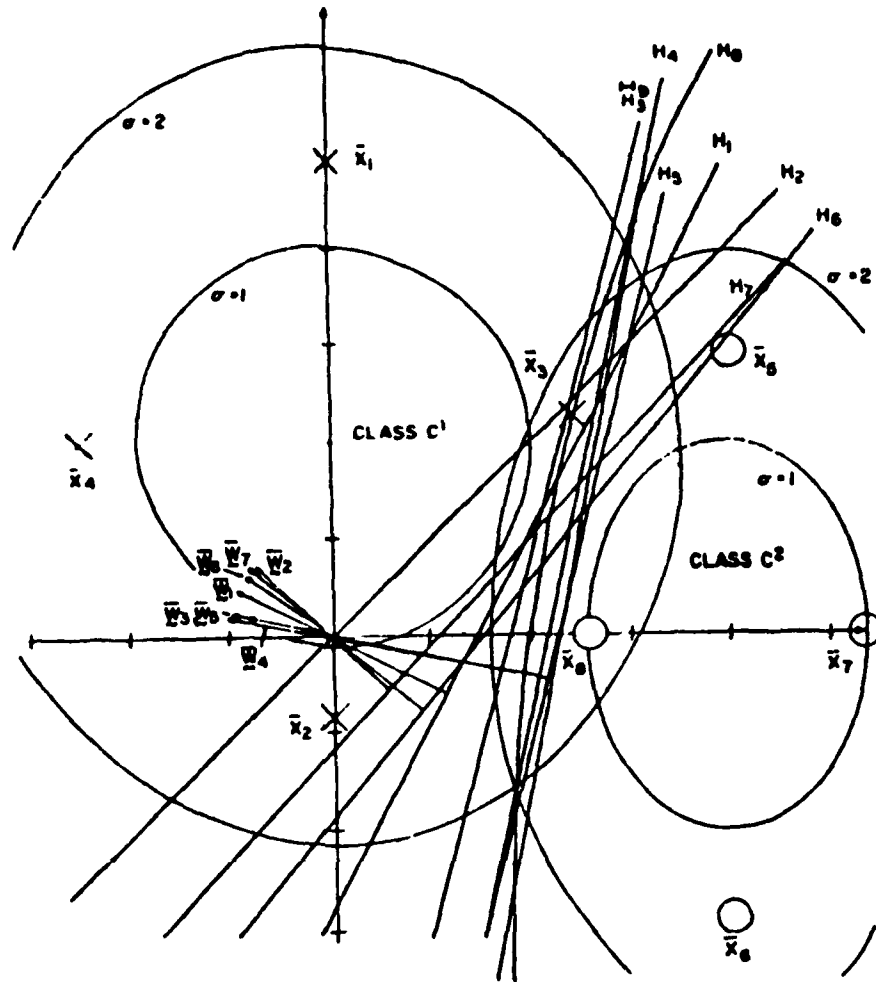
individual latency shifts for each peak. Smoothing was added to make the resulting waveform continuous again. These techniques have been compared (Aunon et al., 1981; McGillem et al., 1985) and it is not surprising that the continuous LCA yielded the more enhanced representation of the ERP.

Wiener filtering (Carlton & Katz, 1980) is probably the most common filtering that has been applied to ERP (McGillem et al., 1981). Minimum mean square error (MMSE) and maximum signal to noise ratio (MSNR) techniques (Aunon & McGillem, 1975; McGillem & Aunon, 1977; Aunon, 1978; McGillem et al., 1981) have also been used. Recently a more powerful class of time varying (Yu & McGillem, 1983) and multichannel time varying filters (McGillem & Aunon, 1985; Westerkamp, 1985) have been utilized.

3.3.3. Classification of ERP

Perhaps the simplest way to classify ERPs is by visual inspection. Unfortunately, while this method may have limited success when applied to the average ERP, it is virtually impossible when applied to a single trial ERP. The basic theory in classification is to develop a feature set which can be used to distinguish between the classes. There are numerous heuristic and statistical manners in which these features can be selected (Fukunaga, 1972; Gevins, 1980; McGillem et al., 1981). These features can be as simple as a present or not present decision or they can be projections on basis functions in an expansion of the signal space. If we view the signal as being the dependent variable, we can map the signal in a space that is spanned by the feature set (Franks, 1981). If each class represents a random process or is the sum of a deterministic signal (e.g. ERP) plus a random process (e.g. EEG), then we will get a mapping similar to the one shown in Figure 3.3.1 (Nagy, 1968). A hyperplane can then be used to separate the classes provided certain requirements are met (Lay, 1982). The hyperplane selected and the manner in which the features are selected define the classification algorithm. This is a very general overview of how classifiers are used and some of the more popular ones are discussed subsequently.

A popular classification technique is Linear Step-wise Discriminant Analysis (LSDA) where features are selected by determining the data vectors for which mean values of each class are most different as measured by a one-way analysis of variance F statistic (Donchin & Herning, 1975). Discriminant functions can also be selected by using the correlation function (correlation classifiers) or the maximum likelihood classifier (Bayes rule) (Sencaj et al., 1979). The Bayes linear classifier will provide optimal results



Common types of linear categorizers x and O indicate the training samples in classes C^1 and C^2 , respectively. The ellipses are the equiprobability contours on the postulated distributions in the test data. The subscripts associated with the hyperplanes and weight vectors pertain to the following categorizers: (1) distance to means, (2) correlation, (3) approximate maximum likelihood, (4) Anderson-Rahadur, (5) discriminant analysis, (6) approximate discriminant analysis, (7) trainable machine, (8) optimal quadratic boundary, and (9) LSDA.

Figure 3.3.1 Linear Classifiers (Nagy 1968)

if all the distribution statistics are normal (Fukunaga, 1972 p. 90). If the covariance matrices for the two classes are not equal then a linear classifier will not provide optimum classification and a two class case will require a quadratic classifier for optimum performance (McGillem et al., 1981). Aunon et al. (1982) gives a comparison of the performance of linear versus quadratic classifiers for ERPs.

One way of selecting features is by using the forward sequential feature selection (FSFS) method. In this procedure the best single feature is selected first using a "training set" of data. The next feature selected is the one that works best in combination with the first one, and so on. This will not yield the best performance, but the computational burden of exhaustively checking the performance of all possible combinations of features can be enormous (Moser, 1984; Halliday et al., 1985).

Other methods of selecting features include Principal Component Analysis (PCA), where features (basis functions) are selected in such a way as to maximize the rate of reduction of the residual variance (Donchin, 1966; Van Rotterdam, 1970; John et al., 1973). This is accomplished by reordering the eigenvectors produced by the covariance matrix in order of decreasing eigenvalues. Critics of this method claim that the resulting components may have little bearing to the original vectors and hence a rotation of the components is recommended (varimax rotation) to force the axes formed by the components to align with a signal vector (Rosler & Manzey, 1981; Wastell, 1981a; Wood & McCarthy, 1984). Childers et al. (1982) utilizes a feature extraction algorithm where the eigenvectors are ordered by decreasing the value of Fisher's ratio instead of simply decreasing order.

3.3.4. Cognitive Uses of ERP

The overriding majority of ERP studies done for cognitive purposes have focused on the effects on the positive peak located approximately 300 msec (P300 or P3) after stimulation (Pritchard, 1981; Ruchkin et al., 1975; Naatanen, 1974; Begleiter et al., 1983; Donchin et al., 1973; McCarthy & Donchin, 1981; Israel et al., 1980). At one point it was believed that P300 was a *measure of the amount of activity of a general purpose cortical processor* (Donchin et al., 1973), however at this point there appears to be a general consensus that P300 is predominantly, if not solely, a measure of "perceptual load". Israel's (1980) classic workload experiment, where P300 was found to vary with loading, utilized a visual monitoring task (large perceptual load) to vary loading. McCarthy & Donchin (1981) found that P300

latency was affected only by stimulus discriminability. Other papers detail the effect of stimulus incentive value (Begleiter et al., 1983) and stimulus probability (Ruchkin et al., 1975). Pritchard's paper (1981) provides an excellent review of P300, while Naatanen (1973) provides a older review of ERP and attention.

A noticeable exception to the P300 fixation has been Gevins. In one experiment (Gevins et al., 1979a), a series of complex cognitive tasks was performed to exercise many different aspects of the information channel. The results were analyzed in a variety of ways and it was found that there was significant change in many of the frequency bands (predominately theta) which correlated with the different tasks. Unfortunately, in a later experiment (Gevins et al., 1979b), where the noncognitive aspects of the task (e.g., limb movement, eye movements, etc.) were more tightly controlled, the results could not be replicated. This illustrates one of the major difficulties with cognitive paradigms: ensuring that the noncognitive aspects of an experiment do not corrupt the data. The paradigm in the proposed research is much simpler than the task performed in these studies and should be less subject to this effect.

3.4. Biomagnetism

3.4.1. Introduction

The study of biomagnetic phenomena is still in its infancy when compared to its older brother, bioelectric phenomena. Figure 3.4.1 (Katila, 1981) shows a comparison between major bioelectric and biomagnetic phenomena, along with their landmark dates. Insight into the current state of research in biomagnetism, as of 1984 (last Biomagnetism Conference), is possible by looking at some statistics regarding the level of research being conducted. Cohen (1985) reports that there are $\simeq 50$ groups worldwide conducting research in biomagnetism, with a total of 200 to 250 researchers including graduate students. Only 50 to 60 refereed papers are published annually, excluding those presented at the conference held every two years. Approximately one third to one half of the papers are brain related. This does not represent a very large worldwide effort for a field that has so much potential. There is still a lot to be done in order for biomagnetism to reach a mature state.

BIOELECTRIC AND BIOMAGNETIC FIELDS				
BIOELECTRIC PHENOMENA	AMPL. (μ V)	BIOMAGNETIC PHENOMENA	AMPL. (pT)	BANDW. (Hz)
ELECTROCARDIOGRAM (ECG) WALLER 1887	1000	MAGNETOCARDIOGRAM (MCG) BAULE ET AL. 1963	50	.05-100
FETAL ELECTROCARD. (FECG) CREMER 1906	5-50	FETAL MAGNETOCARD. (FMCG) KARINEMI ET AL. 1974	1-10	.05-100
ELECTROENCEPHALOGRAPH (EEG) RIEGER 1924	50	MAGNETOENCEPHALOGRAPH (MEG) COHEN 1968	1	.5-30
EVOKED POTENTIALS (VEP) WALTER ET AL. 1946 (SEP) DAWSON ET AL. 1950 (AEP) DAVIS ET AL. 1939	10	EVOKED FIELDS (VEF) COHEN 1975 (SEF) BRENNER ET AL. 1978 (AEF) REITE ET AL. 1978	.1	DC-60
ELECTROMYOGRAM (EMG) ADRIAN 1929	1000	MAGNETOMYOGRAM (MMG) COHEN 1972	10	DC-2000
ELECTRO-OCULOGRAM (EOG) DUBOIS-REYMOND 1849	1000	MAGNETO-OCULOGRAM (MOG) KARP ET AL. 1976	10	DC
ELECTRORETINOGRAM (ERG) HOLMGREN 1865	100	MAGNETORETINOGRAM (MRG) ANTONIEMI ET AL. 1978	.1	.1-30
MAGNETIZATION EFFECTS				
MAGNETIC CONTAMINATION (LUNG) (FC) COHEN 1973				
MAGNETIC SUSCEPTIBILITY PLETHYSMOGRAPHY (MSPG) WIKSWO ET AL. 1974				
HUMAN IRON STORES HARRIS ET AL. 1978				

Earth's Magnetic Field = $70\mu\text{T} \pm 3\text{nT/m}$

Various bioelectric phenomena and their biomagnetic counterparts. Corresponding bioelectric signals do not exist for pure magnetization phenomena. References to the pioneering works of various biomagnetic measurements are made in the reference list from no 1 to no 12. All known biomagnetic fields are not included in the Table.

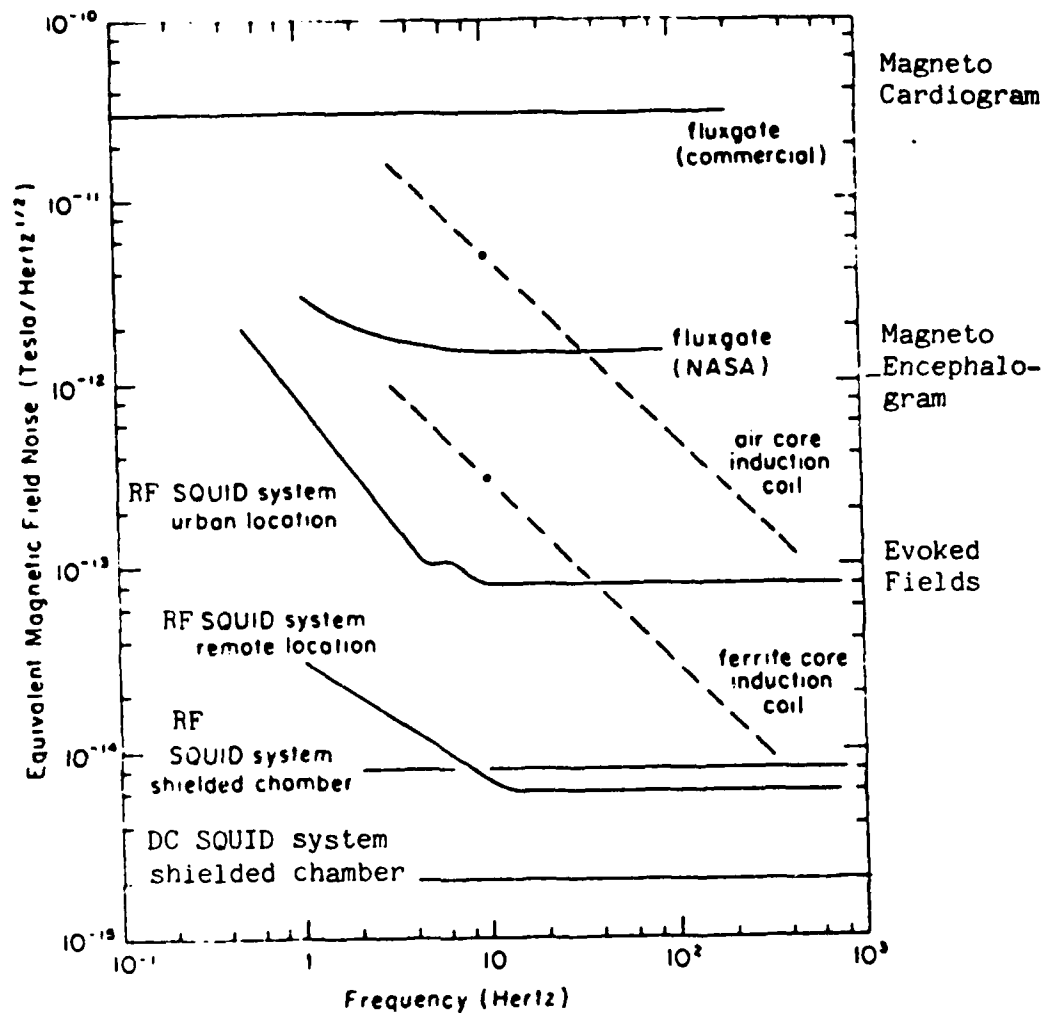
Figure 3.4.1 Bioelectric and Biomagnetic Fields (Katila 1981)

The first recording of the magnetic fields produced by the brain, magnetoencephalogram (MEG), was done in 1968 and took eight minutes of averaging to produce only one cycle of the alpha rhythm (Cohen, 1985). MEG study was not practical, due to the extremely low intensity of the fields, as illustrated in Figure 3.4.2 (Williamson & Kaufman, 1981), until the development of the Super Conducting Quantum Interference Device (SQUID). MEGs are on the order of 1 picotesla (pT), while the brain's magnetic evoked fields (EF) are on the order of 0.1 pT or 100 femtoteslas (fT). The earth's magnetic field is on the order of 70 microteslas (almost eight orders of magnitude greater than EFs), with a spatial gradient of ≈ 3 nanoteslas. Current state of the art measuring systems, DC SQUID in a shielded chamber, have a noise level as low as $15 \text{ fT} / \sqrt{\text{Hz}}$, as measured while collecting data at Wright Patterson Air Force Base (WPAFB). Figure 3.4.3 shows the DC SQUID and support system used at WPAFB, while Figure 3.4.4 shows the shielded room. The channel was bandpass filtered from 0.1 to 25 Hz. This implied a channel noise of 75 fT ($25 \text{ Hz} @ 15 \text{ fT}/\sqrt{\text{Hz}}$), which is not much smaller than the EF.

3.4.2. SQUIDS

A SQUID system is basically a very low noise, very high gain amplifier connected to a sensory loop. The schematic diagram for a DC SQUID system is shown in Figure 3.4.5 from Williamson & Kaufman (1981) and a complete mathematical explanation is available in that reference or Sarwinski (1977). The SQUID itself is a superconducting loop (although the term is often used to imply the entire measurement system) that contains either one (RF SQUID) or two weak links (DC SQUID). A superconducting weak link is a Josephson junction which will oscillate when biased. This oscillation must have an integral number of wavelengths on the loop since it is a closed loop. It can be shown that this requirement translates to an integral number of flux quanta within the loop.

The weak link, Josephson junction, requires a certain amount of bias current to cause this oscillatory effect. The DC or RF notation refers to the manner in which this bias current is supplied. DC SQUIDS use a dc current source as shown in Figure 3.4.5. RF SQUIDS use an RF source which is inductively coupled and hence a larger superconducting loop is needed. This larger loop size is responsible for the greater noise in the RF SQUID. DC SQUIDS became available in the early 1980s and yielded almost an order of magnitude better performance than the RF SQUIDS due to their lower noise



The sensitivity of various magnetic field sensors as indicated by the frequency dependence of the noise level $S_B^{1/2}$, where S_B is the field spectral density.

Figure 3.4.2 Magnetic Sensing Devices (Williamson & Kaufman 1981)

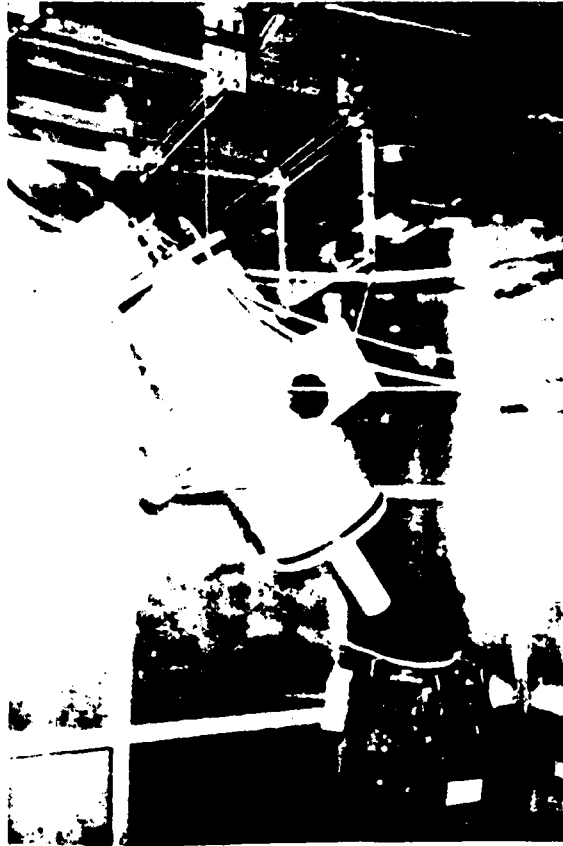


Figure 3.4.3 DC SQUID and Mounting Systems

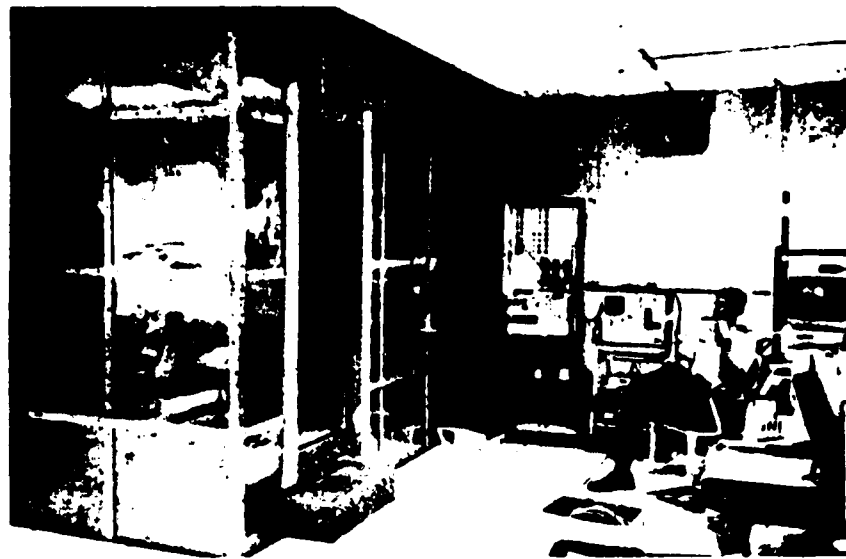


Figure 3.4.4 Shielded Room

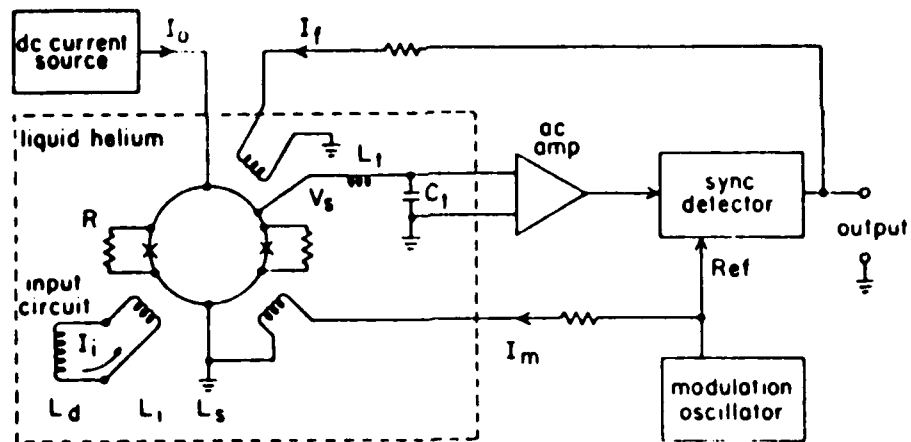


Figure 3.4.5 Schematic for a DC SQUID System (Williamson & Kaufman 1981)

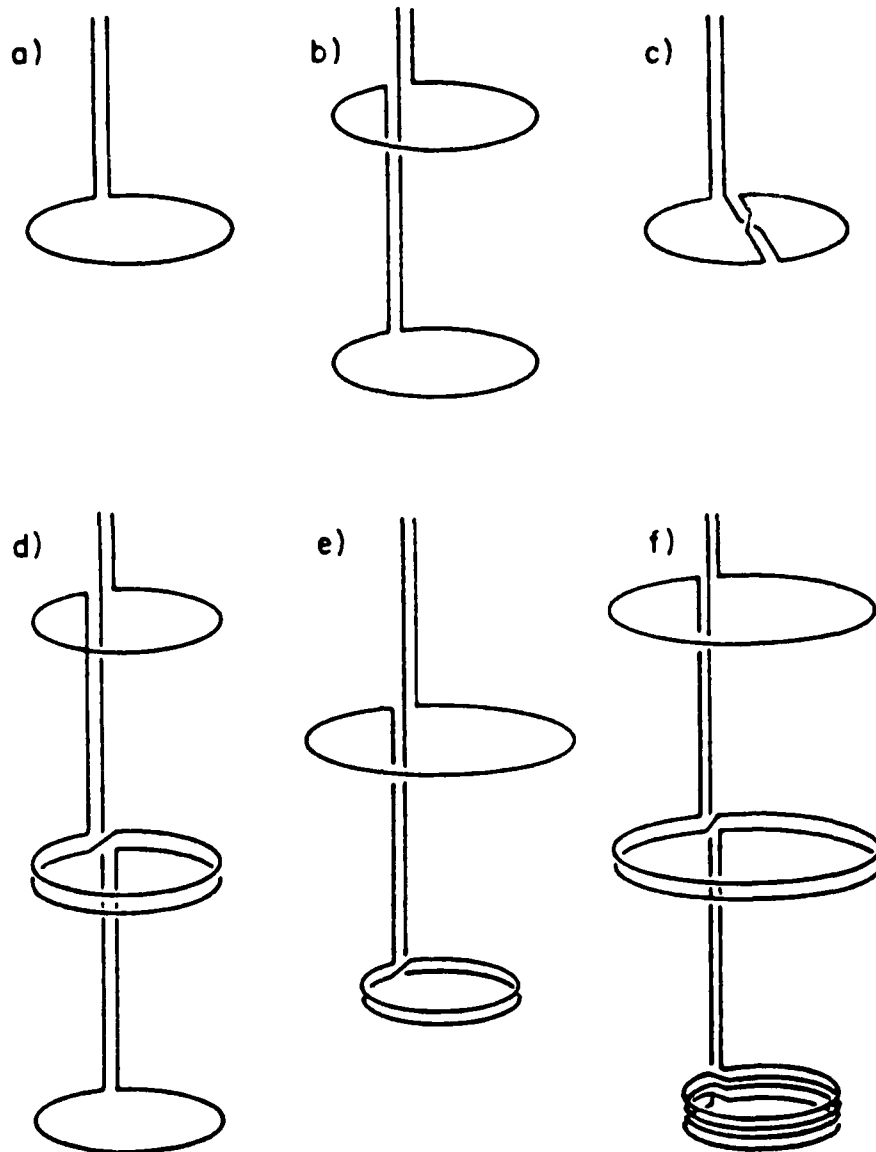
characteristic. Large numbers of trials were necessary with RF SQUIDS in order to record an EF. This number has dropped considerably with the introduction of the DC SQUID (Lewis et al., 1985).

An input signal is inductively coupled with the superconducting loop. As the input signal increases and attempts to add flux to the loop, the superconducting quantum nature of the loop forces an opposing current to ensure that the total flux in the loop remains at the quantum level. Unregulated, this effect would continue until the input signal increased enough to cause the flux in the loop to jump by one quantum level, at which point the opposing current would suddenly drop to zero. The system is regulated through current feedback which is used to keep the loop from jumping quantum levels. The amount of feedback current can then be used as a measure of the input signal. Note that while the input signal is inductively coupled into the squid loop, its origin does not have to be magnetic in nature. SQUID amplifiers have been used with many other types of inputs, but for our concerns, i.e. biomagnetic signals, the input is provided by the detector loop shown as Ld in Figure 3.4.5.

Numerous types of detector coil configurations have been used as shown in Figure 3.4.6. The most popular of these at the moment is the second order gradiometer. This configuration cancels out the effect of constant fields, e.g. earth's magnetic field, and slowly changing fields, e.g. nearby equipment effects. It is the use of this detector loop that allows SQUID measurements to be made in a normal lab environment without shielding. Shielded rooms (Figure 3.4.4) are not necessary, unless you are located in an extremely magnetically noisy environment, such as at WPAFB where the experimental data for this research was collected.

3.4.3. Models for Evoked Fields

In order to understand the nature of the evoked field it is necessary to construct a model as to how it is produced. The most common model used is that of considering the head to be a series of concentric homogeneous spheres and the response to be a simple current dipole. It has been shown for this model that the radial component of the dipole will produce no net external fields (Cuffin & Cohen, 1977) due to the model symmetry. The strength of the external field decreases rapidly as the dipole is located deeper in the brain (Aunon et al., 1984). A dipole located at the center of



Detection coils with the following configurations: a) magnetometer, b) first-order gradiometer, c) off-diagonal gradiometer, d) second-order gradiometer, e) asymmetrical first-order gradiometer, and f) asymmetrical second-order gradiometer.

Figure 3.4.6 Detector Coil Arrangements (Williamson & Kaufman 1981)

the brain would produce no external field since it must be radially oriented. The sharpness of the peaks in the field are much more pronounced for shallow dipoles.

This model also predicts that the radially oriented component of the field will not be affected by the volume conduction associated with the dipole. The tangential components of the field will be affected by the volume flow and consequently they are not often measured. The detector loops shown in Figure 3.4.6 are used for measuring the radial component only and in general, when we refer to the evoked fields we are referring to the radial component only.

The EEG measured on the surface of the scalp represents a spatial smearing of the true activity beneath the skull due to the high resistivity of the skull and the volume conduction flow. The EF is unaffected by the scalp and its radial component should represent a true measure of the activity beneath the skull. This means that the EF responds to a much more localized effect than the ERP and details should show up more clearly. Lewis et al. (1985) was able to report measuring a single trial EF (no averaging) for the first time by placing the detector coil over the localized activity. EFs from shallow sources can be very sharp and only a few centimeters may separate the positive and negative peaks in a field.

The focus of much of the research to date has been solving the inverse problem, i.e. locate the source from the measured field. This has met with considerable success (Aunon et al., 1984; Cohen & Cuffin, 1983). In general the contour mapping of the EF is offset from the mapping produced by the ERP. It has been found that good EFs can be recorded from the sides of the head (Okada et al., 1981; Okada, 1983), while most ERPs measurements are usually recorded from the top of the head. This finding was confirmed while at Wright Patterson and a measurement location for the experiment was chosen to be 7 cm above the right ear. This location was in good accordance with the result obtained by Okada for maximal amplitudes in the 300 to 450 ms latency range.

CHAPTER 4

EXPERIMENTAL PROCEDURE

4.1. Introduction

This experiment basically builds upon the groundwork laid out by Kantowitz (1985b Experiment 1). The time frame for his original work only allowed for a preliminary (phase one) analysis of the data. In the current research, more sophisticated data analysis is performed. Additional electroencephalogram electrodes and the magnetoencephalogram have been added to the data set. There are several advantages to proceeding in this fashion. The experiment is well designed and provides for converging operations. The task differences are fairly simple so that the differences in loading are clearly defined. Several aspects of the workload are varied. The experiment has been shown to give reliable results and results can be compared to the previous research.

4.2. Paradigm

The PRP paradigm presents two stimuli, preceded by a warning tone, in a sequence to form a single trial. First a (W) warning tone (70 dB, 1KHz tone of 500 msec duration) alerts the subject to the start of a new trial. A random foreperiod (W offset to S1 onset) with a mean of 2.0 sec was created by sampling foreperiods of 1.5, 2.0 and 2.5 secs with equal probability. Two stimuli (LEDS) were presented with a fixed inter stimulus interval (ISI) of 62.5 msec and 250 msec. A baseline condition where a second stimuli did not follow the first was also recorded. Figure 4.2.1 shows a diagram of the timing for a single trial.

The LEDS were drawn with equal probability from either a four choice or two choice case. The background illumination was dim, ≤ 0.01 ft-L, and each LED was individually balanced to have an illumination of 0.3 ft-L. A block consisted of 51 such trials and the number of choices and ISI was constant for each block. The first trial from each block was discarded as a preventive measure. The intertrial interval (W onset to W onset) was 6

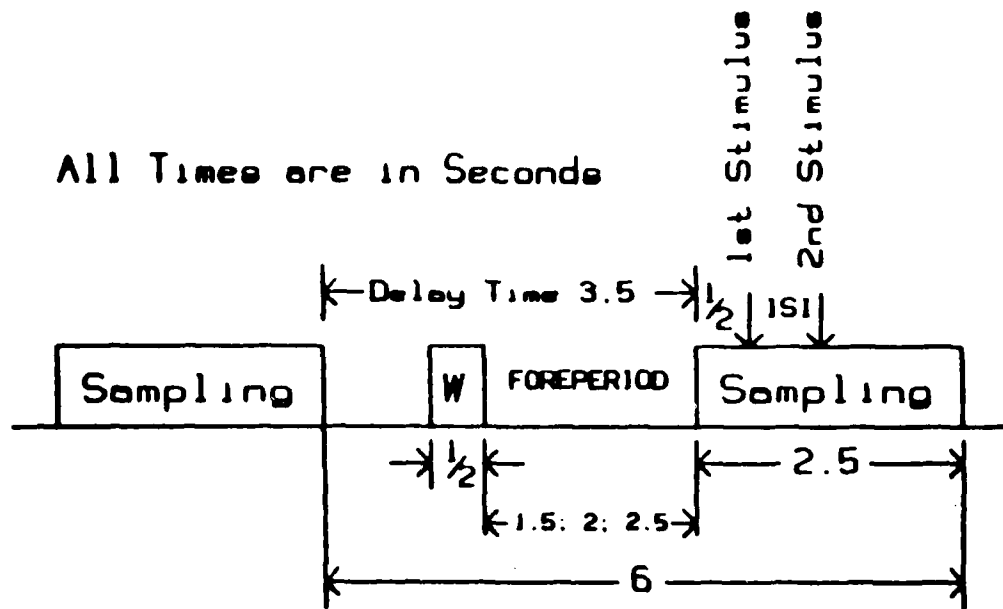


Figure 4.2.1 Single Trial Timing Diagram



Figure 4.2.2 Test Setup with Subject

secs, so that each block lasted ≈ 5 minutes. The subject was informed of the block type (ISI and number of choices) before the start of each block.

The experiment was performed at the Aerospace Medical Research Laboratory (AMRL) at WPAFB due to the availability of a DC SQUID. The subject sat in a shielded room (Figure 4.2.2), viewing a set of 4 horizontally mounted LEDS, slightly below eye level, presented through a small hole in the inner shielded wall. A head support with a strap was used to keep the head steady for the SQUID. The subject viewed the lights from a distance of 122 cm. This was the maximum possible distance due to the size of the room and SQUID support system. The LEDS were placed close together, 4 cm span, so that the entire display was considered foveal.

Responses were made on four piano-like keys, operated by the second and third fingers of each hand. The switches were balanced to have the same approximate characteristic feel and throw. Correct responses extinguished the appropriate LED immediately, while incorrect responses extinguished the LED at the end of the sampling interval for that trial. Each switch could only be used once in a trial. On two choice trials only the inner two lights were used, so that each response was from a different hand. This was not necessarily true for the four choice trials.

A complete experiment (Figure 4.2.3) consisted of a initial relaxation period, a series of 4 blocks, another relaxation period, another series of 4 blocks, and then a final relaxation period. Relaxation periods were 5 minutes long, during which time the doors to the chamber were opened and the strap holding the head in place for the SQUID was released. The number of choices was invariant for each set of 4 blocks, and half of the subjects were assigned to perform the four choice set first, while the other half performed the two choice set first. The 4 blocks within a set consisted of a single stimulation practice block first and the $3!$ orders of ISI and base-line presented in a random order. A total of six lab personnel were used as subjects with one randomly assigned to each order. It took \approx one hour to prepare each subject (attach electrodes, etc) and one and a half hours to perform the experiment. Table 4.2.1 contains a summary of the subjects that were used in this experiment and Table 4.2.2 gives the order of the trial blocks for each subject.

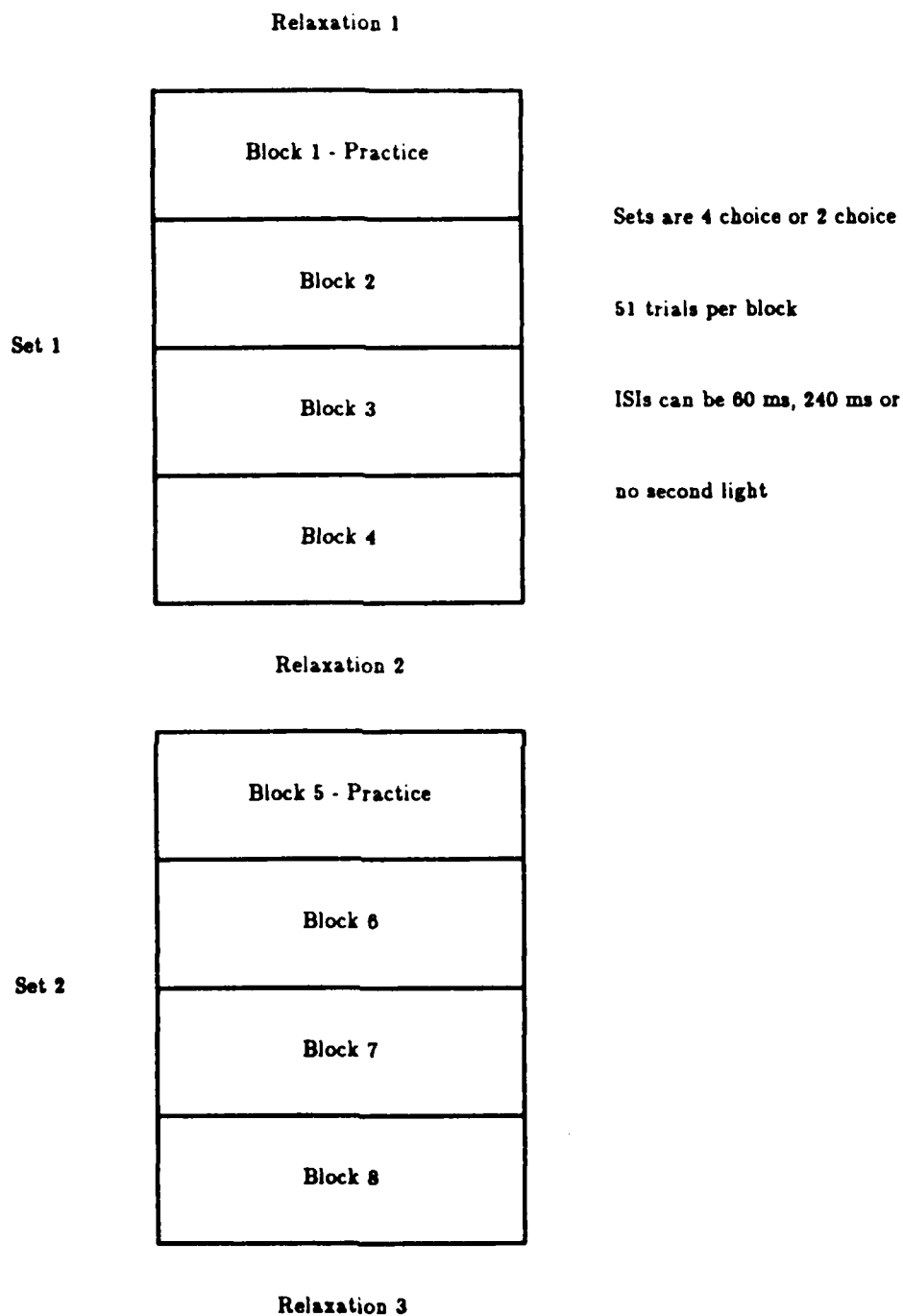


Figure 4.2.3 Experiment Sequence

Table 4.2.1 Subject Summary

SUB 4	WF 35 (lab person) Full disk caused loss of several blocks of data. Lost blocks were rerun to make a COMPLETE DATA SET minus data for the two practice blocks which were not rerun in interest of time.
SUB 5	WF 28 (lab person) COMPLETE DATA SET recorded. Subject was less than willing at times and data is highly corrupted by eyeblinks.
SUB 6	WM 27 (lab person) COMPLETE DATA SET recorded plus 1 additional block with 51 trials of 4 choice 125ms ISI stimulation.
SUB 7	WM 25 (lab person) COMPLETE DATA SET recorded. Subject was well practiced with task before start of this run.
SUB 8	WM 28 (lab person) COMPLETE DATA SET recorded
SUB 9	WM 28 (lab person) COMPLETE DATA SET recorded. Subject was well practiced with task before start of this run. Two additional blocks with 25 trials of 2 choice single stim data were recorded with the SQUID in alternate locations (Files 2_0x & 2_0xx).

Table 4.2.2 Test Sequence for Subjects

FILE #	SUB 4	SUB 5	SUB 6	SUB 7	SUB 8	SUB 9
test00	rest0					
test0	4_0 (48)	rest0	rest0	rest0	rest0	
test1	4_60 (49)	4 (42)	2 (45)	2 (50)	4 (42)	rest0
test2	4_240 (49)	4_0 (48)	2_240 (49)	2_60 (50)	4_240 (48)	2 (45)
test3	rest1	4_240 (11)	2_0 (50)	2_0 (48)	4_60 (42)	2_240 (48)
test4	2_0 (47)	4_60 (7)	2_60 (49)	2_240 (51)	4_0 (50)	2_60 (42)
test5	2_60 (45)	rest1	rest1	rest1	rest1	2_0 (49)
test6	2_240 (49)	2 (36)	4 (49)	4 (50)	2 (47)	
test7	rest2	2_0 (41)	4_240 (50)	4_60 (51)	2_0 (30)	rest1
test8		2_240 (34)	4_0 (49)	4_0 (50)	2_240 (43)	4 (47)
test9		2_60 (32)	4_60 (47)		2_60 (36)	4_240 (43)
test10		rest2		4_240 (50)	rest2	4_60 (35)
test11			rest2	rest2		4_0 (35)
test12				4_120 (49)		rest2
test13						2_0x (22)
test14						2_0xx (25)

Number in () indicates number of valid trials if the entire 2.5 secs of data is to be used.

For each trial, the stimuli (LED numbers), times of stimuli, responses (switch numbers), and times of each response was recorded. Times were recorded to the nearest 0.1 ms relative to the start of the block. It was decided to record all of this information, rather than just the result (correct or incorrect) of each trial as a precautionary move. If unusual results occur for a particular trial, LED or switch it would be beneficial to have this information to see if a pattern was developing.

4.3. Physiological Measurements

4.3.1. Overview

Event related potentials, evoked fields, respiratory and cardiac rate were recorded simultaneously during the experiment. To the best of the author's knowledge, this was the first time that anyone had attempted to record the magnetoencephalogram while simultaneously recording these other physiological measurements. Four channels of EEG data (C3, C4, P3, P4), 1 eyeblink channel (EOG), and 1 MEG channel (EF) were digitized for 2.5 secs (at 240 Hz sampling rate = 600 data pts), commencing 0.5 secs before the first stimulation of each trial. The EF was recorded from a single location 7cm directly above the right ear canal, as described in section 3.4.4. Only the time of each R wave of the ECG and the time of each expiration from the respiration were recorded as these waveforms were not digitized. A Grass Model 79 Polygraph was used to produce a paper recording of C3, C4, P3, P4, EOG, EF, ECG, and respiration.

4.3.2. Respiration & Heart Rate

The ECG was measured from a lead II configuration (Guyton, 1981). The respiration was determined by using an electrolytic strain gauge (Geddes & Baker, 1975 chap. 2) as part of a voltage divider circuit. Both signals were then amplified using Grass 7P511J amplifiers and fed to the data acquisition interface box.

The signals were threshold detected at this point and used to generate system interrupts to record the time of each R wave and the start of expiration. Times were recorded relative to the start of the experiment and to the nearest 0.1 msec. The levels for threshold detection were known to correspond to a specific pen deflection. This facilitated setting the amplifier gains and correlating the paper record to the recorded times and digitized EEG when necessary.

It was important that the plane of the loop formed by the electrolytic strain gauge was kept perpendicular to the plane of the SQUID detector loop. This prevented the magnetic field produced by the current in this loop from being coupled into the SQUID channel. The current was limited to several microamperes and it was found that there was no measurable coupling unless the plane of the strain gauge was parallel to the detector loop.

Respiratory and cardiac rate were also recorded during the rest period between each set of blocks. This was originally done in order to check for changes in baseline. Unfortunately, the subjects were very restless during the rest periods due to discomfort of the head restraint during the stimulus blocks, so these measurements may be biased.

4.3.3 Electroencephalogram and EOG

ERP data was collected, using Beckman miniature electrodes, from the International 10-20 electrode locations (Jasper, 1958) C3, C4, P3, and P4 with the linked mastoids used as the reference electrode. The electroculogram (EOG) was recorded to allow for off line rejection of eyeblink artifact. Grass amplifiers 7P511J were used to amplify four channels of EEG data (P3, P4, C3, C4) and one eyeblink channel (EOG) were amplified using a Grass 7P511J amplifiers and then digitized using the Tecmar labmaster board. These amplifiers as well as the Grass amplifier attached to the SQUID output were calibrated in terms of (A/D units out of the digitizer) / (micro volts in). Offline analysis was done to reject trials due to eyeblink and amplifier saturation.

A waveform analyzer was used to determine the time delay associated with the Grass amplifiers. A DC - 20 Hz linearly swept signal was input into the amplifier and the resulting output was then correlated with the input. The peak of the correlation waveform was used as a measure of the delay time. It was found that the amplifiers had a 2.7 msec delay when set for a .1 to 100 Hz passband and a 4.7 msec delay when the 60 Hz notch filter was added in. Delay times are important since the electroencephalographic data and magnetoencephalographic data are processed by different channels and their digitized waveforms have to be shifted by the difference of their delay times in order to preserve the time synchronization of the measurements.

4.3.4 Magnetoencephalogram

Due to the high level of electrical activity within the building, AMRL had a magnetically shielded room built by EG&G. The room (Figure 3.4.4) was basically a double layer shielded box with the outer layer being a 120 inch cube and the inner layer a 90 inch cube. A wood floor was supported by wooden rods passed through small holes in the bottom of the chamber which provided mechanical isolation from the perma-alloy walls and floor. A double set of sliding doors was used to provide access to the shielded room. When closed, doors were pneumatically held against the frame to ensure electrical contact. A series of electrical feedthroughs were used to send signals in and out of the chamber. Several removable two inch portholes were placed in the walls which allowed ventilation and the use of non-standard cabling. An intercom system was constructed by using a piezoelectric speaker to amplify the sound passed through the chamber walls by a hollow tube.

A simple program, "MUTUAL", written at the Naval Personnel and Development Research Center (NPRDC) in San Diego, California, (Appendix B) was used to compute the mutual inductance between a test loop and the SQUID detector loops. This allowed the computation of the magnetic flux input into the SQUID based on the current in the test loop and was used to verify the SQUID calibration. If the SQUID is functioning properly, the calibration from the test loop should match the manufacturer's specification (Figure B1 in Appendix B). A small loop of wire wrap wire was placed around the tail of the dewar containing the SQUID. The remaining wire was twisted to prevent stray mutual inductance between the test loop and the detector loops.

There are several advantages to using a calibration loop instead of simply relying on the manufacturers calibration. The initial calibration may drift if the alignment of the probe within the dewar changed due to remounting the probe or allowing the dewar to warm. SQUID performance can also deteriorate if ice should buildup under the coil form at the dewar tail. This has happened at NPRDC. Use of the test loop allows a complete operational performance check which would turn up any unsuspected problems.

Another important advantage is that it allows one to compute the entire system transfer function rather than just a part of it. It was found that the delay time from the detector loop to the Grass Amplifier output

was 22.5 msec (SQUID control unit = 3.3 msec, Khronhite 25 Hz Low Pass Filter = 14.5 msec, Grass Amp (with 60 Hz Notch) = 4.7 msec). The difference in time between the SQUID channel and the electrode channels was 19.8 msec (22.5 - 2.7) which is \simeq 5 sample points (at 4.167 ms per point). This time shift has largely been ignored by other researchers, though it can be quite significant depending on the amount of low pass filtering that is added to the SQUID channel. The filtering used at NPRDC amounted to \simeq 30 msec of delay which would make peak alignment between the EEG and MEG quite difficult if this went uncorrected.

Any metal that the subjects were wearing or carrying was first removed before they entered the shielded chamber. The subjects wore a nylon cap over the electrodes and a circle was drawn on this cap to aid in SQUID placement. The subjects sat with their head strapped into a padded head rest at a \simeq 45 degree angle. The SQUID was lowered into place until it just made contact with the cap and then was backed off slightly (see Figure 4.2.2). Since the shielded room and SQUID support at WPAFB had not been used for experimentation before, it was necessary to try several types of subject placement before deciding on the final arrangement.

Due to the distraction and time it took to open and close the pneumatically sealed doors, the doors were only opened during the rest periods between the sets of 4 blocks and not between each block. This meant the subject was sealed in the room for periods of \simeq 30 minutes until the doors were opened during the rest periods. The ventilation system was inadequate to keep the chamber well ventilated over such a long period of time. The seating and head support were also not adequately comfortable for the long durations inside the chamber. One must consider that there is some refinement period after the construction of a new system before it can reach it's optimal performance.

4.4. Data Acquisition System

An extremely powerful, yet flexible data acquisition system (Figure 4.4.1) was built for this experiment and to serve the future needs of the EEG laboratory at Purdue University. The system was built around an IBM AT, equipped with internal and external hard disk drives and a Tecmar Labmaster data acquisition board. A Grass Model 79, 8 channel polygraph and EEG amplifier was used to amplify and make paper recordings of the signals. A complete description of the system along with all the applicable software and documentation is given in Appendix C.

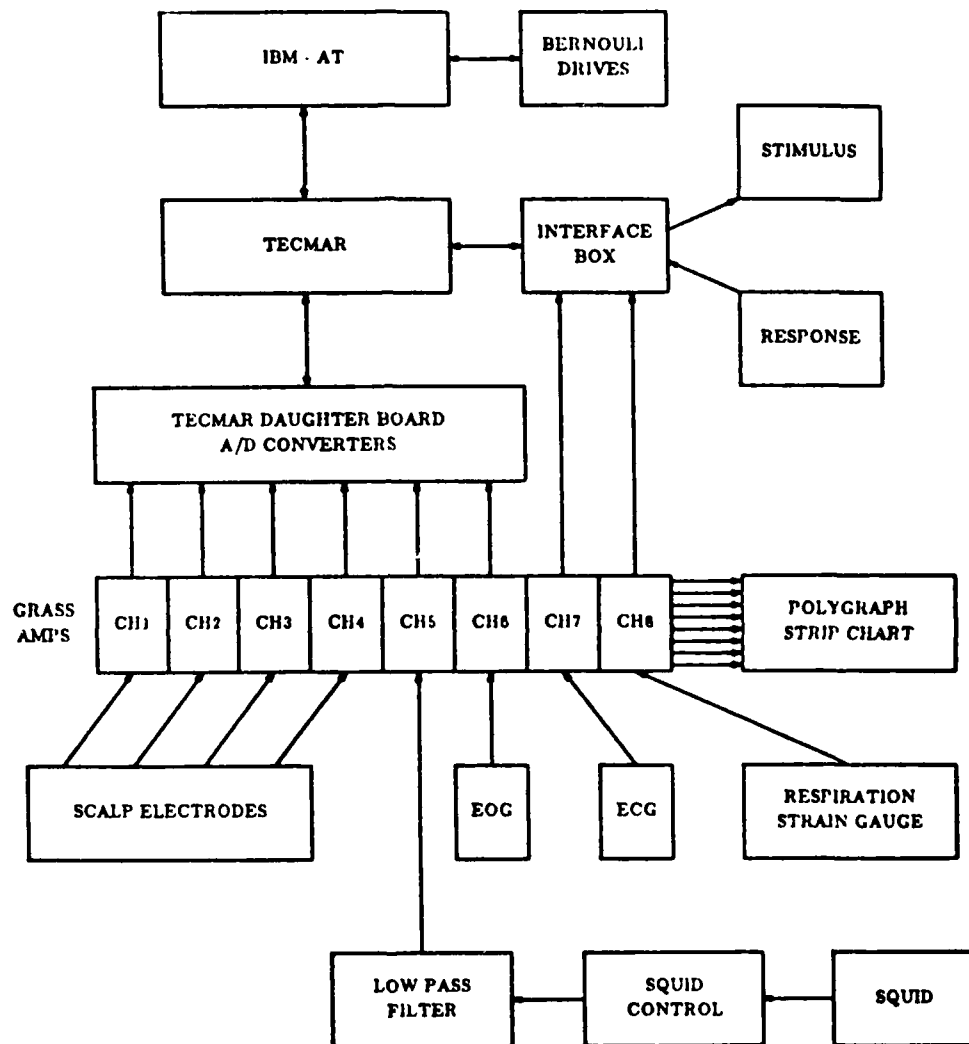


Figure 4.4.1 Equipment Configuration

An interface box was made connecting the IBM and Tecmar Labmaster to the outside world. This interface brought all of the Tecmar data and control lines to a standard printed circuit (PC) board edge connector. The interface also supplied power and made 22 output lines available via a choice of three standard types of connectors mounted on the outside of the box. All of the necessary hardware for this experiment was then built on a single PC board and inserted in the box. The advantage of this type of system is that its flexible nature allows many different experiments to be built without worrying about all the connections. Each new experiment is simply constructed on a PC board and all the necessary connections are already provided.

The data acquisition program was written primarily with the intent of being used for real time processing, however this experiment did not demand real time processing. The start of each trial is initiated by the controlling program and the entire data collection, stimulus presentation and response recording is interrupt driven. A flag is raised at the end of the trial so that the controlling program knows to start the next task. The controlling program is free to handle other task while data is being collected. The advantage of this method is that only 5% of the systems resources are used in the data acquisition and 95% of the resources are available for real time processing. The data acquisition code is extremely flexible and is able to handle a variety of tasks (e.g., turning off the light if the correct switch is pushed) in a background mode that can respond to both hardware and software interrupts.

All data collection is double buffered. This means that while the interrupt driven code is collecting the data for the current trial, all results from the previous trial are available to the controlling program. The controlling program can process that data and a decision can be made as to whether that data should be stored. The interrupt driven code checks to see if a reject flag has been raised by the controlling program before storing the data on the next trial, hence the name double buffered. The time frame for this research did not allow the implementation of real time error checking, however all of the mechanisms for adding this checking to the package are already in place. All of the design problems were worked out at Purdue and then the entire system was taken to WPAFB, to incorporate their SQUID into the design.

CHAPTER 5

PRELIMINARY DATA ANALYSIS

5.1. Introduction

This chapter discusses the procedures and results from the preliminary data analysis phase. The EEG and MEG channels are screened for artifact rejection and then filtered. Average responses and power spectral estimates from the prestimulus interval are computed. Mean inter-beat-intervals for the heart and two measures of heart rate variability, sample variance and mean square successive difference, are computed. Statistics based on subject performance are also computed. Data from six subjects are used in this phase of the analysis.

5.2. Behavioral Responses

The response performance is used to provide the converging criteria called for by Kantowitz (1985a). The "RESPOND" program was used to compute the response performance. For each trial, the response fell into one of six categories:

- 1) The subject did not respond to the presented stimuli.
- 2) The subject responded before the stimulus was presented.
- 3) Both responses were wrong.
- 4) First response was wrong, second response was correct.
- 5) First response was correct, second response was wrong.
- 6) Both responses were correct.

The program counted the number of trials that fell into each of these categories and computed the average response times and standard deviations for the last four categories.

In this report, the term "correct response" shall be taken to mean that both responses were correct unless specifically stated otherwise. Since the first trial was always rejected as practice, there was a maximum of 50

possible correct responses. We shall only consider the response time statistics for the "both correct" case since the majority of the responses fall in this category and incorrect responses do not appear to have the same mean as correct responses. $\overline{RT1}$, $\overline{RT2}$ represent the average response times and σ_{RT1} , σ_{RT2} represent the standard deviations of the response times to the first and second stimulus respectively. In comparing average response times between loading conditions for a given subject, it is important to consider the standard deviations of the response times. In Figure 5.2.1, bar graphs are used to compare response times to the first stimulus. The upper part of each bar is shaded to indicate the range corresponding to the average plus and minus the standard deviation. The midpoint of the shading is the average response time for that loading. Figure 5.2.2 is a similar graph for the response to the second stimulus. Figure 5.2.3 is a multiple bar graph which shows the percentage of correct responses to both stimuli for all subjects at all loading levels. In most cases the accuracy was in excess of 90%, i.e. more than 45 out of 50 correct responses to both stimuli.

One of the ways that loading can be determined is from the task description. The goal of this research is to find behavioral and physiological correlates to the predicted increase in loading. We can evaluate the task descriptions (i.e. number of choices and ISI) in light of the hybrid capacity model to assign relative loadings. There are 15 possible binary comparisons for the six loading conditions, however not all of the outcomes can be predicted from the model and task description. The model cannot provide any insight as to the relative loading between a 2 choice 62.5 msec ISI loading and a 4 Choice 250 ms ISI for example. Table 5.2.1 summarizes the predicted outcomes for the behavioral variables for six of these binary comparisons based on the model and task description. Three of these comparisons measure the effect of choice by holding the ISI (or single stimulation) constant. Two of these comparisons measure the effect of ISI on double stimulation by holding choice constant. The last binary comparison is between the highest (4 Choice 62.5 msec ISI) and lowest (2 Choice 250 msec ISI) loading for the double stimulation cases. Note that the hybrid capacity model does not specifically require a slower response time to the first stimulus when the ISI is decreased from 250 msec to 62.5 msec, however it does allow for this to occur while some of the previously discussed models do not.

Table 5.2.2 summarizes the measured outcomes of these comparisons across all six subjects. Mean response times to both stimuli for each subject

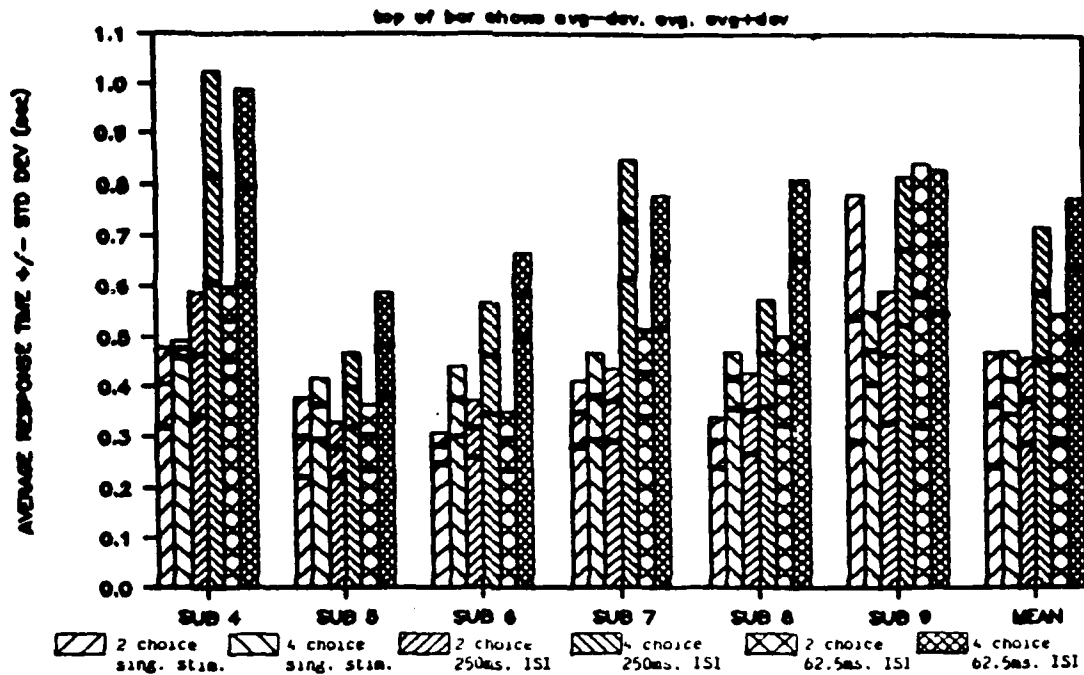


Figure 5.2.1 Response Time to 1st Stimulus (RT1)

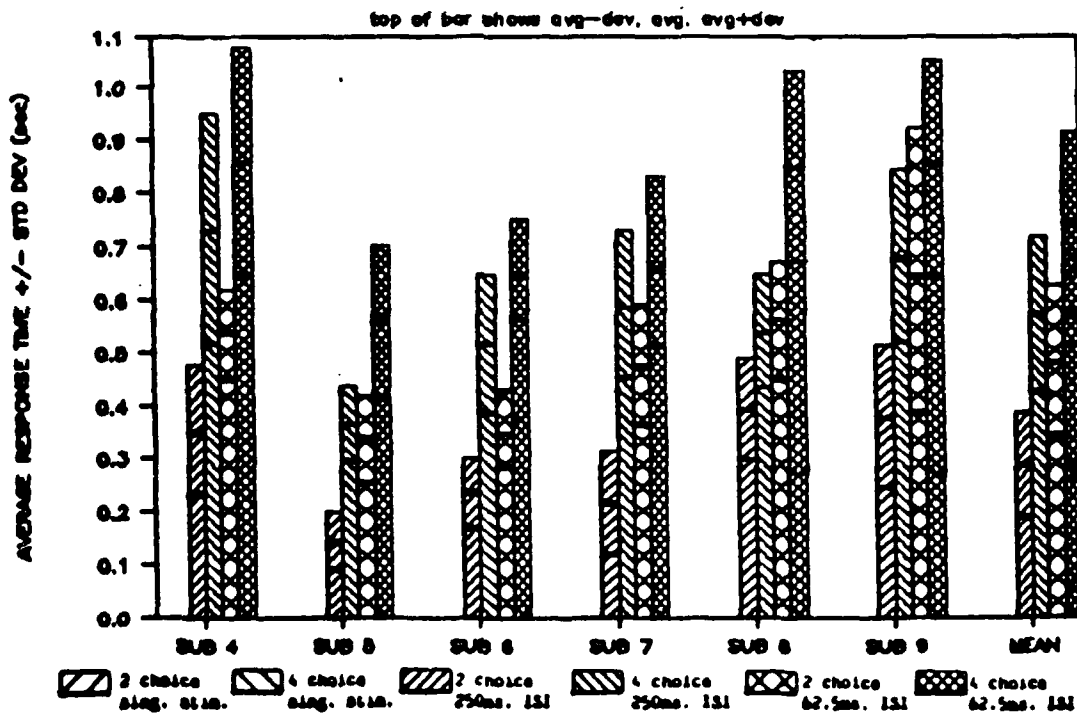


Figure 5.2.2 Response Time to 2nd Stimulus (RT2)

were compared for the different loading cases by using a modified student t-test which allowed for different variances and sample sizes (Sachs p. 271, 1984). Only the trials with correct responses were used in computing the mean and variance for the t-test. The response time entries in Table 5.2.2 indicate those subjects whose mean response times were in agreement (at the 0.025 level of significance) with the outcome predicted in Table 5.2.1. A means test could not be performed on the accuracy since scoring for each trial was not based on a numerical value. The accuracy entries in Table 5.2.2 are a simple comparison of the mean accuracies for each subject. One would expect that response time 2 and accuracy would be the best indicator of loading for the double stimulation cases and that response time 1 and accuracy would be the best indicators for the single stimulation control cases. It appears that at the higher levels of loading (columns 3,5,6), the measured performance is consistent with the predicted performance. At the lowest loading level (column 1), the measured performance is not consistent across the subjects for either accuracy or response time.

Our task description and behavioral measures have resulted in the same effective loadings for the higher loading comparisons. What remains is for the physiological measures to also converge on the same relative loadings. This would provide the converging criteria that are called for by Kantowitz (1985a).

5.3. Heart Rate

The mean IBI and two measures of its variance were computed in this phase of the analysis. The data acquisition program, "DAC", recorded the time of each R wave as determined by threshold detecting the ECG. It was necessary to first prescreen this data to check for ectopic beats and shifts in baseline which might cause a P wave to cross the threshold. The "INTBEAT" program was used to produce two output files, one containing the actual time of each beat and the other containing the inter-beat-intervals (IBI). The program would also flag beats whenever the IBI was not within an acceptable range (500 - 1500 ms). Plots of the IBI were then made for each loading case and examined visually for unusual deviations that the computer program may have missed. The statistics for the IBIs were then computed from the corrected data.

Mean heart IBI ranged from a low of 757 msec (subject 5) to a high of 1218 msec (subject 9) as computed over the 5 minute interval that it took to complete one trial block. The threshold detector was set at $\approx 30\%$ of the R

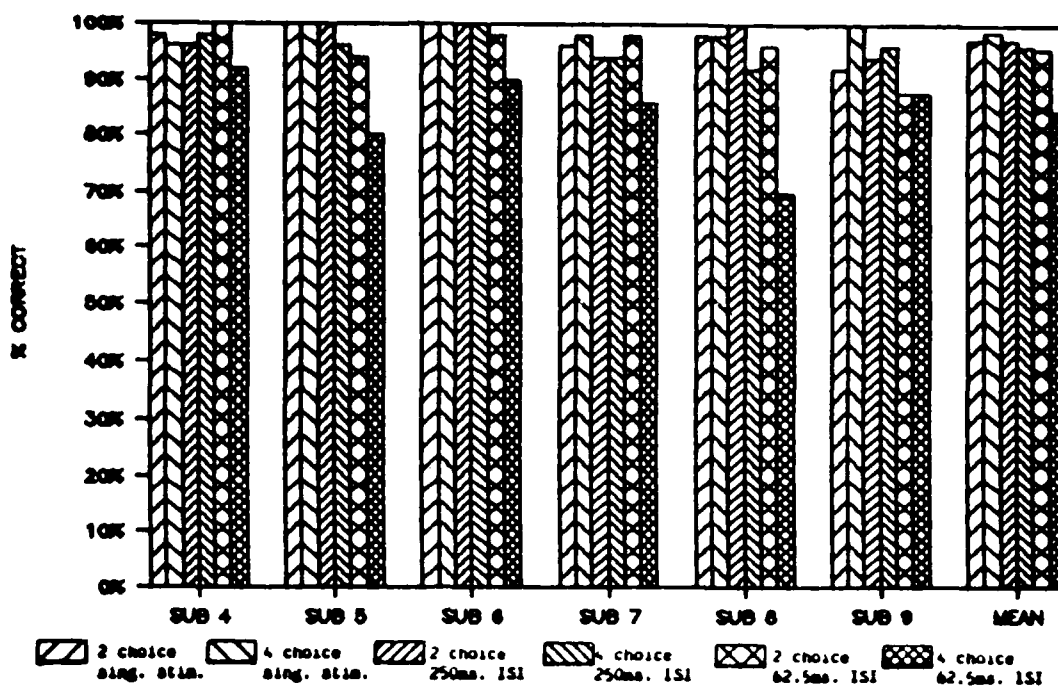


Figure 5.2.3 Percent of Correct Responses to Both Stimuli

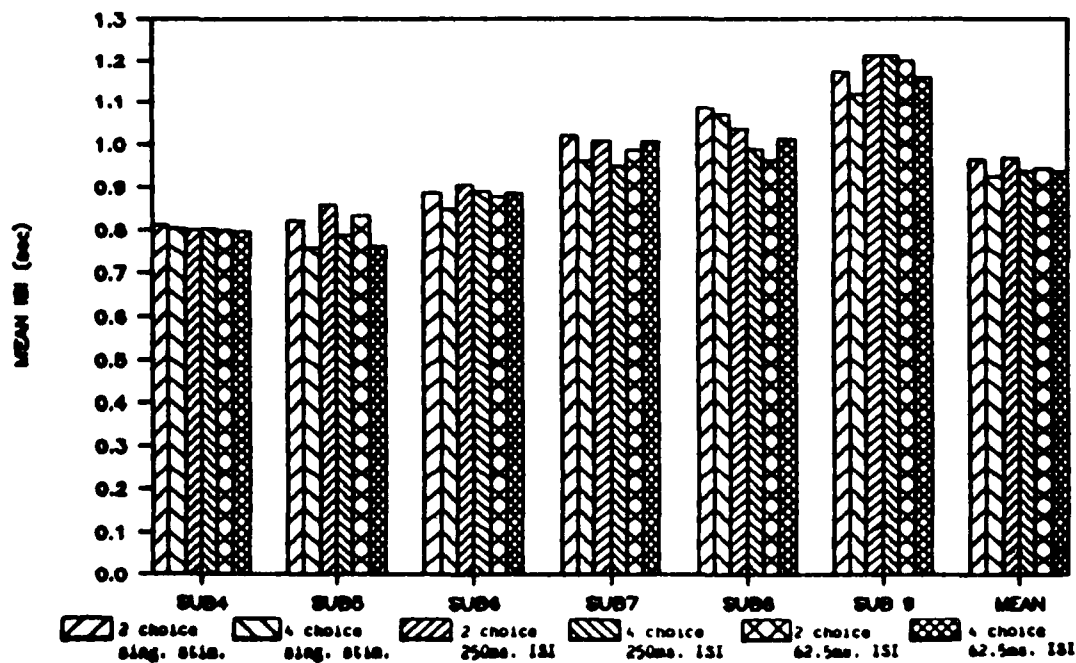


Figure 5.3.1 Mean Heart Inter-Beat-Interval (IBI)

Table 5.2.1 Behavioral Performance Predicted by Task Description

	4 Choice Single Stim vs 2 Choice Single Stim	4 Choice 250ms ISI vs 2 Choice 250ms ISI	4 Choice 62.5ms ISI vs 2 Choice 62.5ms ISI	2 Choice 62.5ms ISI vs 2 Choice 250ms ISI	4 Choice 62.5ms ISI vs 4 Choice 250ms ISI	4 Choice 62.5ms ISI vs 2 Choice 250 ms ISI
Accuracy	Lower	Lower	Lower	Lower	Lower	Lower
RT1	Slower	Slower	Slower	Slower	Slower	Slower
RT2	-	Slower	Slower	Slower	Slower	Slower
Loading	Higher	Higher	Higher	Higher	Higher	Higher

Table 5.2.2 Measured Behavioral Performance

	4 Choice Single Stim vs 2 Choice Single Stim	4 Choice 250ms ISI vs 2 Choice 250ms ISI	4 Choice 62.5ms ISI vs 2 Choice 62.5ms ISI	2 Choice 62.5ms ISI vs 2 Choice 250ms ISI	4 Choice 62.5ms ISI vs 4 Choice 250ms ISI	4 Choice 62.5ms ISI vs 2 Choice 250 ms ISI
Accuracy	4,5,6,8	5,6,7,8	All	5,6,8,9	All	All
RT1	4,5,6,7,8	All	All	4,5,7,8,9	5,6,7	All
RT2	-	All	All	All	All	All

wave amplitude, to favor an extraneous threshold crossing (e.g. P wave crossing) over a missed R wave. Only one recorded IBI was found to exceed 1500 msec and the paper recording of the ECG showed it to be due to a long beat. It is extremely unlikely that there were any missed beats due to the setting of the threshold level and that the INTBEAT program and visual screening of the data did not indicate any.

There were several instances where a recorded IBI was measured below 500 ms. These questionable beats were evaluated on an individual basis by examining the paper record of the ECG. Ectopic beats which inhibited the

next regular beat were simply moved to the mid point between the two adjacent beats. This can be justified since ectopic beats are of local origin to the cardiac tissue and are not regulated by the autonomic mechanisms. P wave crossings were removed entirely. Only one P wave crossing (across all six subjects) was found during the testing blocks, though several others were found during the rest periods when the subjects tended to move around more. The only ectopic beats found were for subject six, who had 5 ectopic beats out of a total of ≈ 2000 beats.

The "HR_AVAR" program was used to compute the mean inter-beat-interval, \overline{IBI} , and two measure of the heart rate variability, s^2 and $\delta^2/2$ from the corrected IBIs. s^2 is the sample variance which provides an unbiased estimate of the population variance, σ^2_{IBI} . $\delta^2/2$ is $1/2$ mean square successive difference (MSSD) and also provides an unbiased estimate of σ^2_{IBI} . These values were computed in the following fashion:

$$\overline{IBI} = \frac{1}{n} \sum_{i=1}^n IBI_i \quad s^2 = \frac{\sum_{i=1}^n IBI_i^2 - (\overline{IBI})^2/n}{n-1} \quad \delta^2/2 = \frac{\sum_{i=1}^{n-1} (IBI_{i+1} - IBI_i)^2}{2(n-1)}$$

It has been claimed that $\delta^2/2$ is a better estimate of heart rate variability (HRV) than s^2 (Heslegrave et al, 1979; Sharit and Salvendy, 1982) since the former removes the effects of nonrandom gradual trends.

Figure 5.3.1 shows the \overline{IBI} for all subjects and loading cases. There does not appear to be any consistent pattern of \overline{IBI} with loading. The \overline{IBI} remains constant for some subjects, increases with loading for other subjects and decreases with loading for others. Figure 5.3.2 shows the sample standard deviation, $\sqrt{s^2}$ and Figure 5.3.3 shows the standard deviation computed from the MSSD method, $\sqrt{\delta^2/2}$. Again, no consistent pattern emerges to distinguish the workload. Note that $\delta^2/2$ is consistently smaller than s^2 , with the exception of subject 9 four choice loading conditions. It appears that there may be trends present which inflate the value of s^2 .

s^2 should match $\delta^2/2$ exactly for independent samples from a normally distributed population. The ratio $\delta^2/2 / s^2$ can be used as a test of independence for normal data (Sachs p. 373, 1984). If we assumed 50 sample points for each estimate, then this ratio would have to be less than .772 to reject the hypothesis of independence at the .05 level of significance. This would correspond to a threshold of .878 ($\sqrt{.772}$) for the standard deviation estimates which are shown in Figures 5.3.2 and 5.3.3. A quick visual inspection

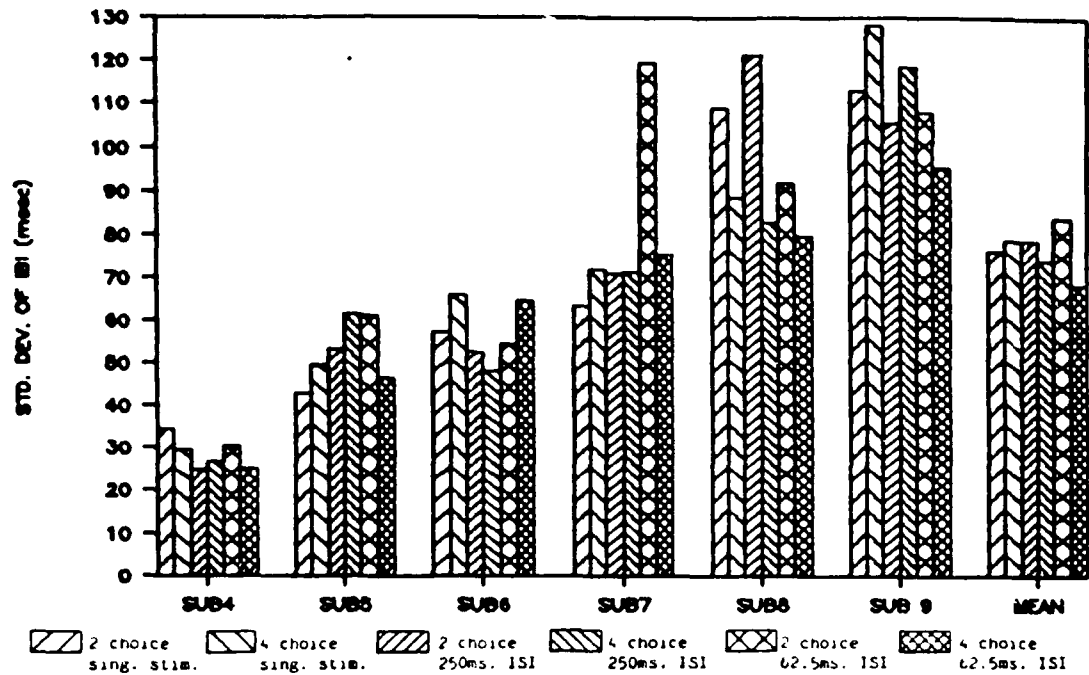


Figure 5.3.2 Standard Deviation of Heart IBI

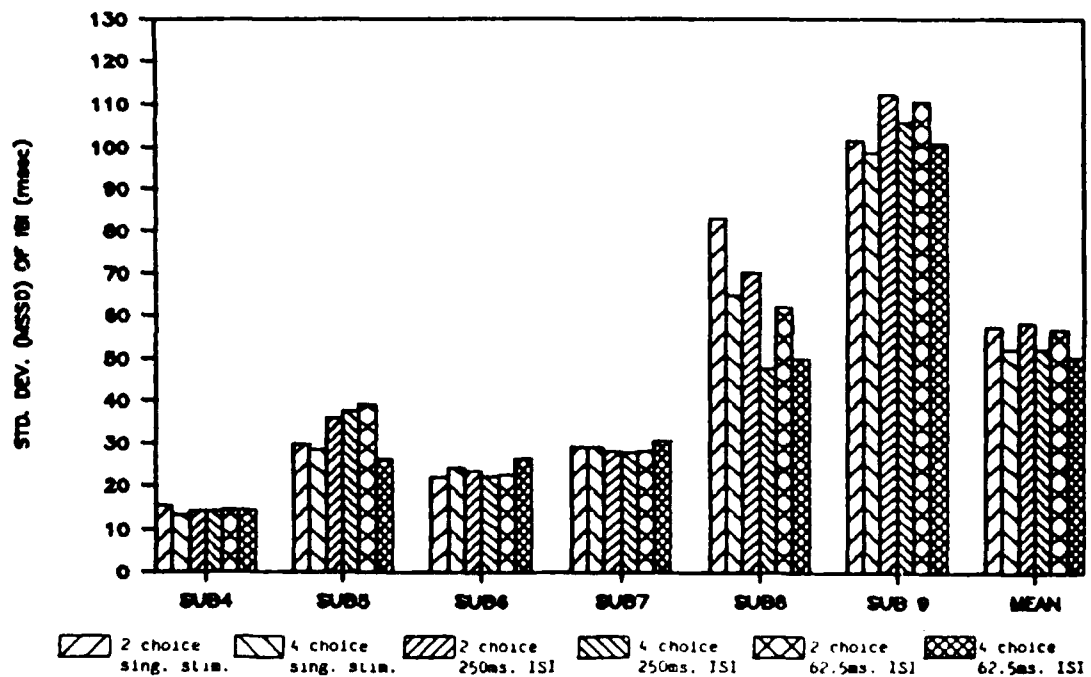


Figure 5.3.3 Standard Deviation of Heart IBI (MSSD Method)

of these two graphs show that the IBIs are not independent with the possible exception of subject 9. A standard run test (Sachs p. 375, 1984) was performed on the IBIs and the hypothesis of independence was rejected at the .0001 level of significance for all cases. This is to be expected since the heart rate is a controlled process.

One of the largest factors in HRV is due to respiratory sinus arrhythmia (SA). The IBI should stretch with each expiration and shrink with each inspiration. Figure 5.3.4 shows a plot of IBI over a 100 sec time interval during the middle of the 4 choice - 62.5 msec loading block for Subject 7. The start of each expiration is indicated on this graph by a "*". There appears to be a trend of increasing IBI following the start of expiration. Note that not every expiration will cause the IBI to increase since there are many other factors that also increase IBI. SA will be dealt with more in the advanced analysis section.

Figure 5.3.5 shows IBI over the course of an entire block (same subject and block as above) along with response time 1 (RT1) plotted and aligned on the same time scale. It does not appear that RT1 is correlated with IBI. Also the IBI starts at 750 msec and quickly rises to its mean value 1000 msec about which it varies for the remainder of the block. This pattern of a sudden increase in IBI was not consistent across other blocks though it did appear in some.

5.4. Respiration

The respiration channel was processed in a similar fashion to the heart rate channel. The data was first prescreened by the computer to flag all breaths that did not fall within between 1.8 secs and 8 secs. These breaths were corrected by examining the paper record. The resulting data was then graphed for all of the cases and was visually inspected for abnormal variations that might need correction.

The chest circumference was the variable that was measured as an indicator of respiratory activity. Unfortunately this measurement does not necessarily make monotonic transistions from inspiration to expiration so simple threshold detection does not produce the most reliable estimate of respiratory rate. A significant problem was partial breaths which did not always cross the threshold level. Careful selection of the amplifier gain and threshold level for each subject reduced the number of beats that had to be resolved by visual inspection to approximately 1% for subjects 5,6,7 and 8,

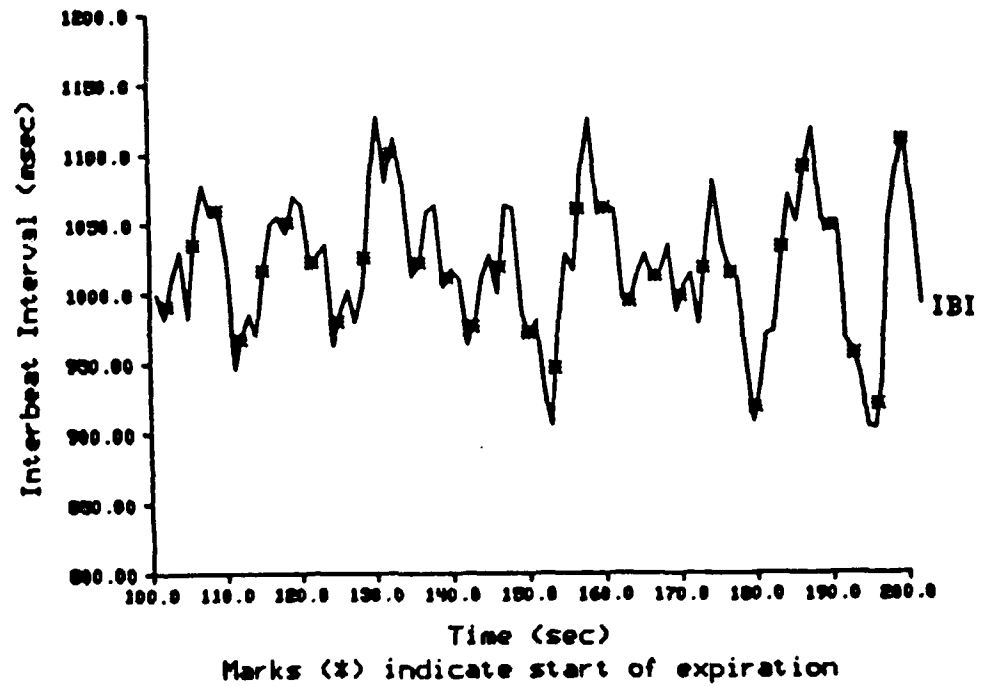


Figure 5.3.4 Respiratory Sinus Arrhythmia as Detected using IBI

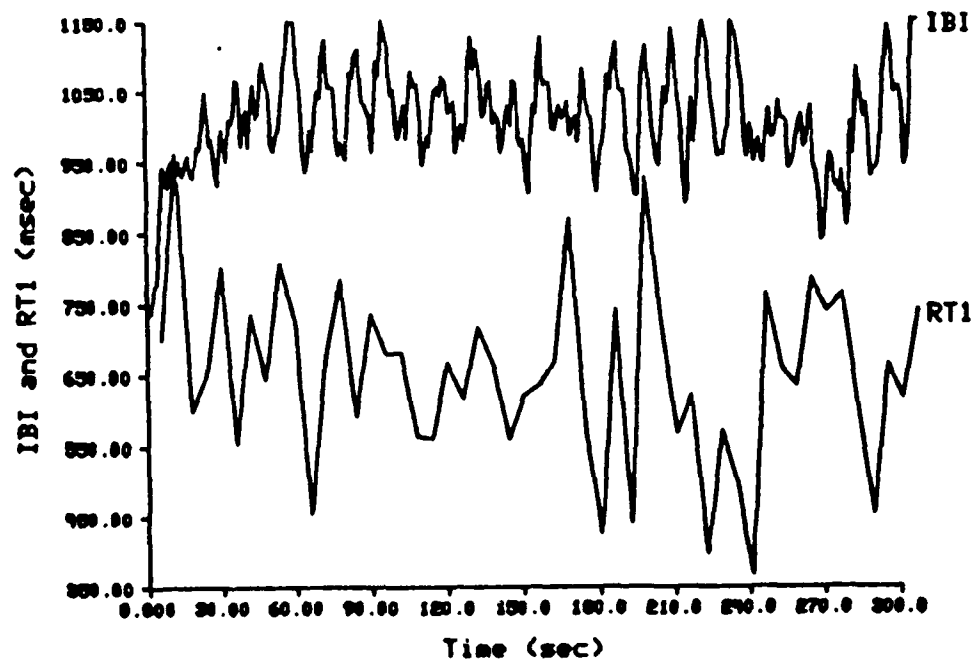


Figure 5.3.5 IBI and RT1 over one Complete Block

but was at a 10% level for subjects 4 and 9. Subject 4 liked to hold her breath and on several occasions this exceeded 14 seconds in length. Subject 9 liked to take partial breaths which did not cross the threshold level. In hand correcting the data, the rule that was used was that a partial breath had to be at least half the size of the surrounding breaths in order to be counted.

Figure 5.4.1 shows the mean respiratory intervals for the corrected data. Figure 5.4.2 shows the standard deviation of the respiratory intervals. Note the large standard deviation for Subject 4 due long periods of holding her breath. At the start of this research it was believed that it was only necessary to record the respiratory rate so that its contribution to the power spectrum of the heart rate could be accounted for. In retrospect it would have been advantageous to digitize the respiratory waveform so that the correlation function and depth of breathing could be considered.

5.5. Artifact Rejection and Prefiltering

The "PERFORM" program is used to determine records to be rejected. It produces two output files, a reject file and a performance file. The reject file is a list of records to be rejected along with a code number which indicates the reason for the rejection. The performance file contains the subject response times along with a performance code which indicates the subjects accuracy for each of the non-rejected trials. The "ANALYZE.H" file contains a list of these codes.

A trial was rejected due to any of the following reasons:

- 1) If it was the first trial.
- 2) If an eyeblink occurred during the response.
- 3) If channel saturation occurred during the response.
- 4) If the subject did not respond to the stimulus.
- 5) If the subject responded before the stimulus was presented.

An eyeblink is defined as a change exceeding 75 microvolts on the electrocuculogram (EOG) channel during a 100 msec interval. Saturation is defined by any of the channels (EEG, MEG or EOG) exceeding the input range of the digitizer for more than 2 sample pts (≈ 8 msec). Table 4.2.2 shows the number of valid trials for each block if the entire 2.5 secs of data was

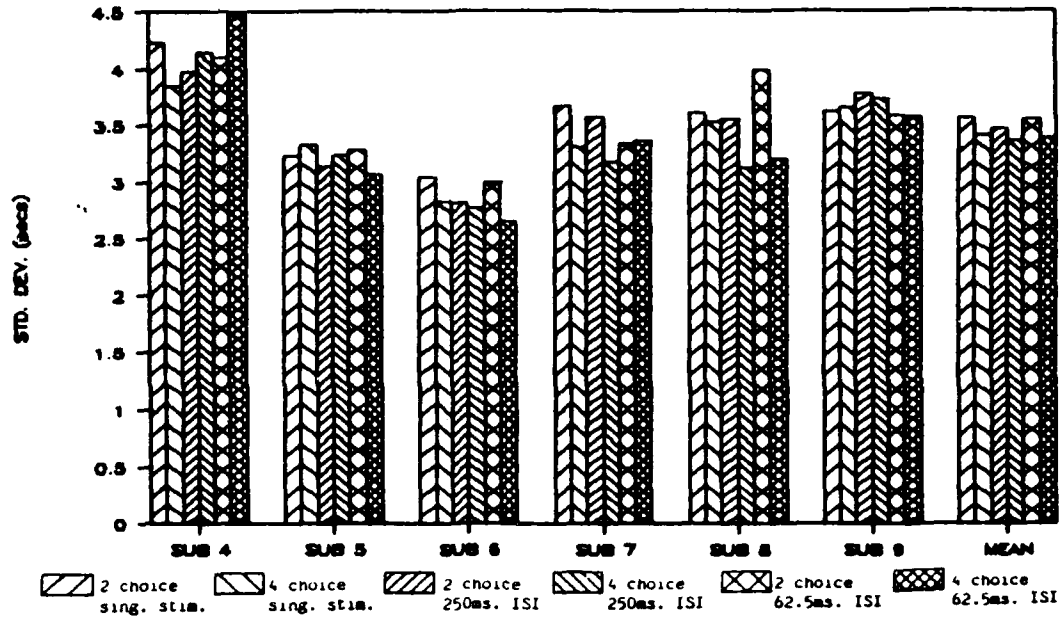


Figure 5.4.1 Mean Respiratory Intervals

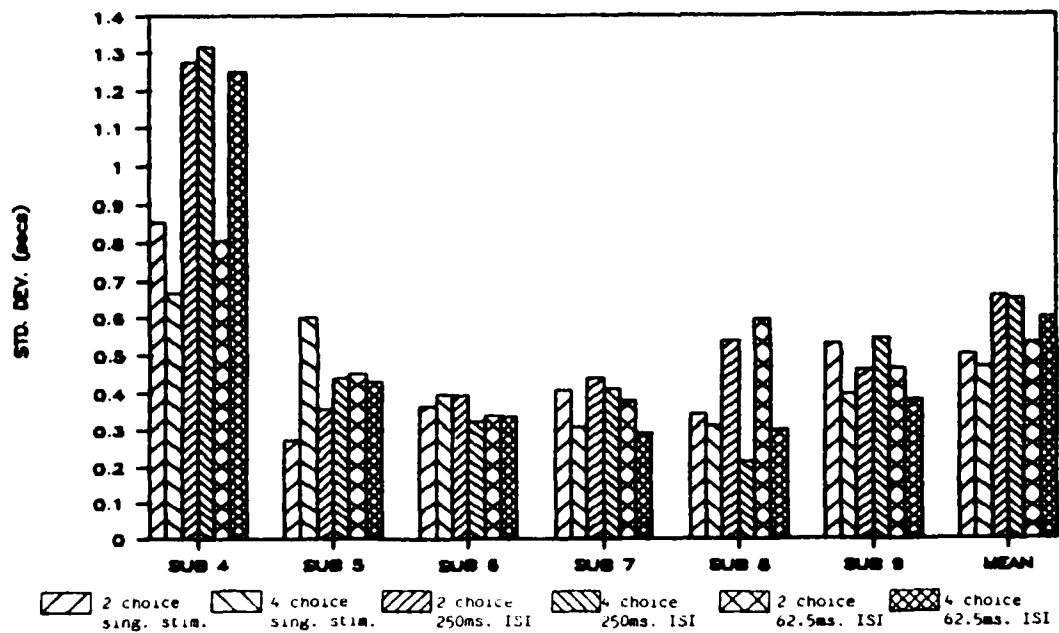


Figure 5.4.2 Standard Deviation Respiratory Intervals

searched for artifacts. In each stage of the analysis, the PERFORM program was rerun to reject only those artifacts that occurred within the time intervals that would be used in that analysis.

The "SEP_CHAN" program is used to separate and filter the individual channels from the original data file which was recorded by the "DAC" program. The program reads the reject file so that only valid records are separated out and used in computing the average. The program has the ability to undersample the data and also produces an average of the filtered data for each channel.

The data for each channel is first calibrated and the linear trend that is formed by joining the first and last point is then removed. This trend is removed to force the first and last point to zero and remove step effects when the filter is applied. Note that this is different from the "best fit line" formed by a linear regression of all the data points. A lowpass filter designed using the Remez Exchange Algorithm (McClellan et al 1979) is then applied. The filter used was a 39 point finite impulse response filter with a 3 dB cutoff at 25 Hz. The passband ended at 21 Hz (.075 dB ripple) and the stopband started at 30 Hz (36 dB attenuation). The data was then undersampled by selecting every fourth data point. This reduced the effective sampling rate to 60 samples per second and the corresponding nyquist rate to 30 Hz.

5.6. Power Spectral Estimate of the Prestimulus Interval

For each trial, there was a $\frac{1}{2}$ sec prestimulus interval which consisted of 30 data points after undersampling. The "BT" program was used to compute the power spectral estimate (PSE) from this interval. The mean of each trial was removed and the autocorrelation function (equivalent to the autocovariance for a zero mean process) was then computed. The individual autocorrelations, R_{xx} , were then averaged to produce an average autocorrelation, \bar{R}_{xx} . \bar{R}_{xx} was then normalized by dividing by $\bar{R}_{xx}(0)$. A 30 point Hamming window was applied and a discrete symmetric cosine transform was taken to produce a normalized ($\sigma^2=1$) estimate of the power spectrum (PSE) with a frequency spacing of 2 Hz.

If we consider the ongoing EEG (i.e. prestimulus interval) to be a random process, then $R_{xx}(0)$ is an estimate of the power, $\hat{\sigma}^2$, in this process. This estimate has a mean, $\hat{\sigma}^2 = \bar{R}_{xx}(0)$, and a standard deviation, $\sigma_{\hat{\sigma}^2} = \sigma_{R_{xx}(0)}$. $\bar{R}_{xx}(0)$ is the scaling factor for the PSE and $\sigma_{R_{xx}(0)}$ is the

standard deviation of the scaling factor. Appendix A contains graphs of all of the normalized power spectra obtained in this fashion. There are 30 graphs (6 subjects x 5 channels) with 6 levels of loading per page to allow easy comparisons of the PSEs. $\bar{R}_{xx}(0)$ is indicated on these graphs as the "avg power" and $\sigma_{R_{xx}(0)}$ is indicated as "std dev".

In general, two cases can be separated by a classifier if the difference between the means of the estimator, $m_1 - m_2$, is significantly greater than the sum of the standard deviations $\sigma_1 + \sigma_2$ (Figure 3.3.1). Since $\sigma_{R_{xx}(0)}$ is typically on the order of $\bar{R}_{xx}(0)$, the power in the prestimulus interval would not provide a good measure of the mental loading. If any particular frequency (or frequency band) of the PSE was to be used as a classifier, then the respective PSEs would have to be markedly different due to the large variance of the scaling factors. Looking at the graphs in Appendix A, this does not appear to be the case. It may still be possible however to use the PSE as an indicator of workload by comparing the ratio of power in different bands, i.e. $PSE_1(2\text{Hz}) / PSE_1(8\text{Hz})$.vs. $PSE_2(2\text{Hz}) / PSE_2(8\text{Hz})$. Due to the large amounts of data, this will require significant analysis before trends may appear and is not covered in this phase.

5.7. Average of the ERP and the EF

Appendix A also contains 30 graphs (six subjects x five channels) of the average low pass filtered ERPs and EF with six loading levels per page. The time scale for these graphs was chosen to show the entire data record ; from 500 msec before the first stimulus was presented to 2000 msec after it. In all discussions and graphs of ERPs and EFs in this report, the time of the first stimulus is indicated as time = 0, with the prestimulus interval indicated by negative time. Recall that the first step in the data analysis was to check for artifacts. In appendix A, the entire data records (-500 msec to 2000 msec) were checked for artifacts, so that only artifact free records were used when computing the average.

There are advantages and disadvantages to checking the entire record for artifacts. The purpose of the graphs in appendix A was to show a baseline condition (prestimulus), a response (ERP) and then a return to baseline. A single unremoved eyeblink anywhere in the region of interest can have a detrimental effect on the average. The danger in rejecting trials from an artifact anywhere in the record is that you may reject more of the data than you want. Subject 5 was someone who might be described as an

uncooperative subject. In the high level loading case of 4 Choice - 62.5 msec ISI, there were only 6 unrejected trials out of a possible 50. If the interval for checking artifacts was reduced to a 1 second interval (0 to 1000 msec), the number of valid trials increases to 36. If the region of interest was only the response (0 to 1000 msec), 36 valid trials would yield much better results than 6. Throughout the data analysis, trials are rejected only if the artifact falls within the region of interest for the current analysis. Trial rejections due to artifacts (-500 to 2000 msec range) ranged from an average high of 23 rejections per block (50 trials) for Subject 5, to less than 1 per block for Subject 7, with Subject 6's data being artifact free.

Average responses are normally computed by aligning the stimulus of each trial and computing an average. The peaks in the ERP and EF which are synchronized to the stimulus become enhanced by this average. What about peaks that may be synchronized with the response instead of the stimulus? The "AVER" program was used to compute unfiltered averages which were aligned with the stimulus, first button push (response) and the second button push. Figure 5.7.1 shows a conventional average (stimulus aligned) overlaid on an average computed by aligning the first response. The response aligned average was computed by shifting each trial so that it had an effective response time of 0 msec, i.e. the first response was shifted to the first stimulus. In plotting the graph, the conventional average was shifted forward by the average response time to the first stimulus and the x-axis was scaled accordingly so that 0 msec still represented the time of the first stimulus. Figure 5.7.1 is taken from the P3 channel of Subject 6 with a 2 Choice - 240 msec ISI loading. The average response time for the first stimulus (\overline{RT}_1) was 314 msec with a standard deviation (σ_{RT1}) of 58 msec. The response aligned average appears to have a slightly higher peak in the 575 msec range and shows a negative and positive peak in the 800 msec range which do not appear in the conventional average. It is too soon to draw any general conclusion from this during the preliminary analysis stage.

Figure 5.7.2 is identical to Figure 5.7.1 except that the EF (MEG) channel was used instead of P3. It was hypothesized that the oscillations in the conventional average, centered about 900 msec, were due to head movement. The oscillations in the response aligned average have the same form but are $\approx 50\%$ larger in amplitude. The MEG has been shown to be relatively impervious to eyeblink contamination due to its localized measurement area. It is therefore unlikely that this oscillation is a muscle artifact from the button push, since the arm was resting on a table and the finger only

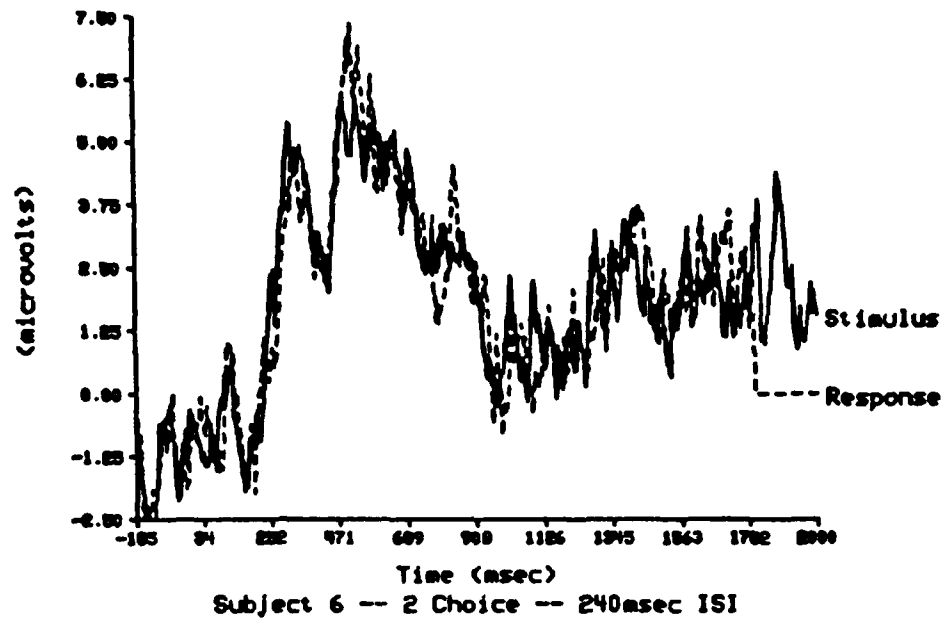


Figure 5.7.1 Stimulus Aligned (solid) & Response Aligned (dash):
Average P3

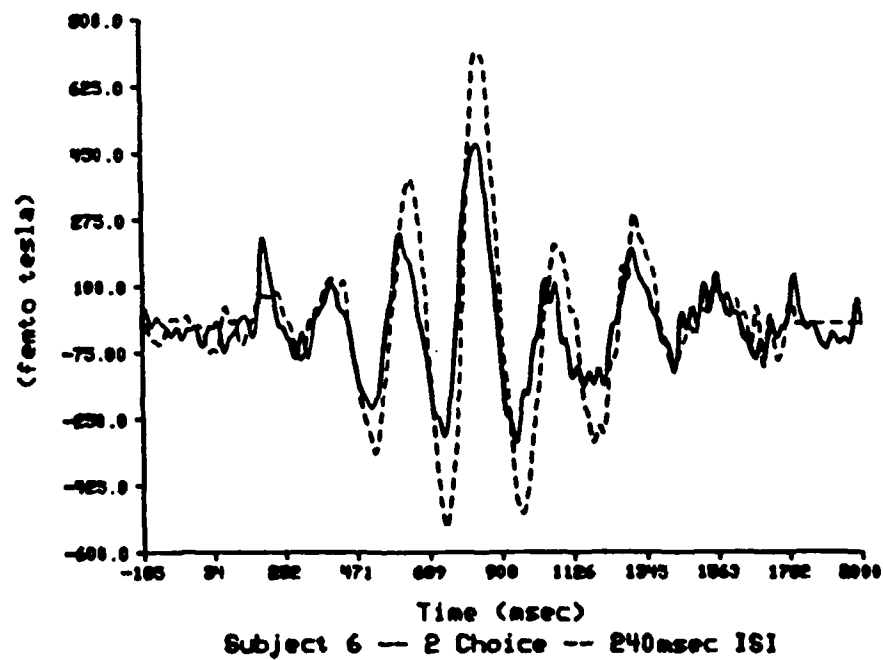


Figure 5.7.2 Stimulus Aligned (solid) & Response Aligned (dash):
Average Meg

had to move a small amount. It appears likely that in a rush to respond (speed was stressed over accuracy) a subject might cause their upper body to oscillate slightly when hitting the buttons. Recall that the buttons were large piano like keys which only required a small light throw since they were attached to microswitches. The subject rested his fingers on the keys, but the sheer size of the keys invited the subject to use all his force when pushing down on the keys.

In addition to possible head movement when a subject "pounced" on the response keys, the head restraint could not keep the head properly positioned under the SQUID with a high degree of precision. The SQUID was placed at the beginning of each set and checked at the end of each set (4 blocks and ≈ 30 min. later). The worst case movements were measured to be a 2 cm lateral shift and a 1 cm in-line shift (i.e. dewar tail was 1 cm from the scalp). Since it took several minutes to open and close the doors to the shielded room, it was not practical to check the SQUID at the end of each block. There appear to be oscillations in many of the EFs and it does not appear at this point that any useful information can be obtained from these measurements. Time constraints on this research did not allow the opportunity to wait for a more optimal support structure which would have reduced this effect.

CHAPTER 6

ADVANCED ANALYSIS

6.1. Introduction

This chapter discusses the procedures and results from the advanced data analysis stage. Spectral techniques are used in analyzing the heart IBI. The latency corrected average is used to further analyze the ERP and identify significant peaks in the response. These peaks are then checked for correlation with the behavioral measures of response. All six subjects are used in the heart rate analysis, but only three subjects are used for the latency corrected averaging due to the computational effort.

6.2. Spectral Estimates of Heart Rate Variability

In the previous chapter, heart rate variability was measured in terms of the mean inter-beat interval, $\overline{\text{IBI}}$ and two estimates of the variance of the beat interval, σ^2_{IBI} . In this chapter we break down the heart rate variability into its spectral components to get a clearer picture of the effect of loading. Several methods of computing the Power Spectral Estimate (PSE) were tried and compared before deciding to use the PSE computed from the autocorrelation of the Low Pass Filtered Event Series (LPFES).

There are two main ways of computing the the spectral content for a series of point events. The spectrum of intervals is based on computing the power spectrum of the inter-beat intervals using the beat number as a transform axis while the spectrum of counts is based on using time as a transform axis (Dekwaadsteniet, 1982; Sayers, 1973, 1975). In computing the spectrum of counts, one can either work with a series of unevenly spaced delta functions (R-waves) or interpolate to an evenly spaced time series of inter-beat intervals. Mohn (1975) discusses the computational complexity of the various methods. Rompelman et al. (1982) advocates interpolating by rounding off the location of the R-wave to the nearest msec to preserve the binary nature of the signal and then computing the PSE by use of a sparse discrete fourier transform. Kantowitz (1985b) simply performed a linear

interpolation from IBI to instantaneous heart rate at 1 sec spacing. The signal can also be interpolated by passing the delta functions corresponding to the R-waves through an ideal low pass filter, which amounts to a Sinc function ($\text{Sin}(x)/x$) interpolation. The French-Holden algorithm (Peterka et al., 1978). is a computationally efficient means of applying an ideal low pass filter with a frequency cutoff at $\frac{1}{2}$ the desired sampling rate by utilizing the binary nature of the series of point events. The output of this filtering has been called the low-pass filtered event series (LPFES). DeBoer et al. (1984) has shown that for small variations from the mean, the spectrum of intervals and the spectrum of counts will yield very similar results if all factors are taken into account.

One problem when computing the PSE from the IBIs is that the data from each block does not have the same number of points. Each data block consist of ≈ 311 seconds of data which may contain anywhere from a low of 254 heart beats to a high of 404 heart beats depending on the subject and loading. The first approach investigated was to compute the spectrum of intervals for the first and last 128 beats in each trial in an attempt to see if the spectrum was changing with time. This was computed by taking the magnitude of the Fast Fourier Transform (FFT) of these two data segments after multiplying by a Hamming window and removing the mean. While the two PSEs both identified peaks in the same location, the variance in the estimates were too large to draw any specific conclusions. For normally distributed data, the variance of the estimate of the PSE computed in this fashion can be as large as the estimate itself. To reduce this variance, the data is usually segmented and the PSEs from each segment are averaged to compute the periodogram (Kay & Marple, 1981). This reduction in variance is obtained at the cost of reducing the frequency resolution.

There are still several limitations with proceeding in this fashion. The number of data points for each case are still not equal so that the processing and variance for each estimate will be slightly different. If it is desired to view the PSE on a frequency (Hz) axis by scaling the spectrum of intervals axis by $1/\overline{\text{IBI}}$, then each case will have a different delta frequency (df), the measurement between frequency components, and a point by point comparison between the cases may not be accurate. We can not break the data into many segments since we need a resolution fine enough to distinguish between the energy in the thermal band (0.025 Hz) and the blood pressure band (0.07 - 0.1 Hz) as shown in Figure 3.1.1. A frequency increment of 0.01 Hz in the PSE would correspond to a 100 point data segment if the $\overline{\text{IBI}}$ was

equal to 1 sec. If the processes which mediate the heart rate are functions of time rather than beat number, their energy in the transform domain will be spread out due to the nonuniform sampling of a periodic signal.

The spectrum of counts approach was deemed to have too many shortcomings and was abandoned after a cursory look did not indicate any consistent differences between the loading cases or the PSE from the first and last 128 beats in a trial. It was decided to pursue a spectrum of counts approach based on the French-Holden algorithm to provide the interpolation. The output of the French-Holden algorithm was inverted to convert it from a rate to a set of interpolated equal spaced IBIs which was then used as the LPFES. The first 10 and last 10 points were discarded to remove interpolation effects and any initial change in heart rate with the start of a block. The mean was then removed from the data. The autocorrelation was then computed and windowed by a Hamming window to 64 positive lag values. The Fourier transform of the windowed autocorrelation function was then used as the PSE for analyzing the heart rate variability. This is the classical Blackman Tukey PSE (Kay & Marple, 1981).

In order to fully understand the effects of the interpolation, the PSE was computed using the Blackman Tukey method from both the original IBI data and the equally time spaced LPFES. Figure 6.2.1 shows the original IBI series superimposed on the LPFES series at two different time scales for Subject 6 at a 2 Choice - 250 ms ISI loading. The original data is plotted as the IBI versus the time of each beat, while the LPFES series is simply plotted at 1 sec increments. The 60 sec time scale in Figure 6.2.1a shows that it takes several data points for the interpolation to settle down which is why the first and last 10 points were discarded when computing the PSE. The 300 sec time scale in Figure 6.2.1b shows that the interpolation algorithm is capable of following the data for the duration of a trial.

Figure 6.2.2 is a graph of the Blackman Tukey PSE of the original IBI series overlayed with the Blackman Tukey PSE of the LPFES when the time increment (dt) was set to the mean IBI for the same data as shown in Figure 6.2.1. No normalization was performed on either the x or y axis since, according to DeBoer et al. (1984), these two graphs should be equivalent for small coefficients of variation (σ/mean) of the inter-beat interval. The mean IBI for this data was 906 ms while the standard deviation was 53 ms. A time increment of 906 ms in the French-Holden algorithm implies a cutoff frequency of 0.55 Hz ($1/(2 \times 906 \text{ ms})$). Scaling the frequency axis of the spectrum of intervals PSE according to the method outlined by DeBoer

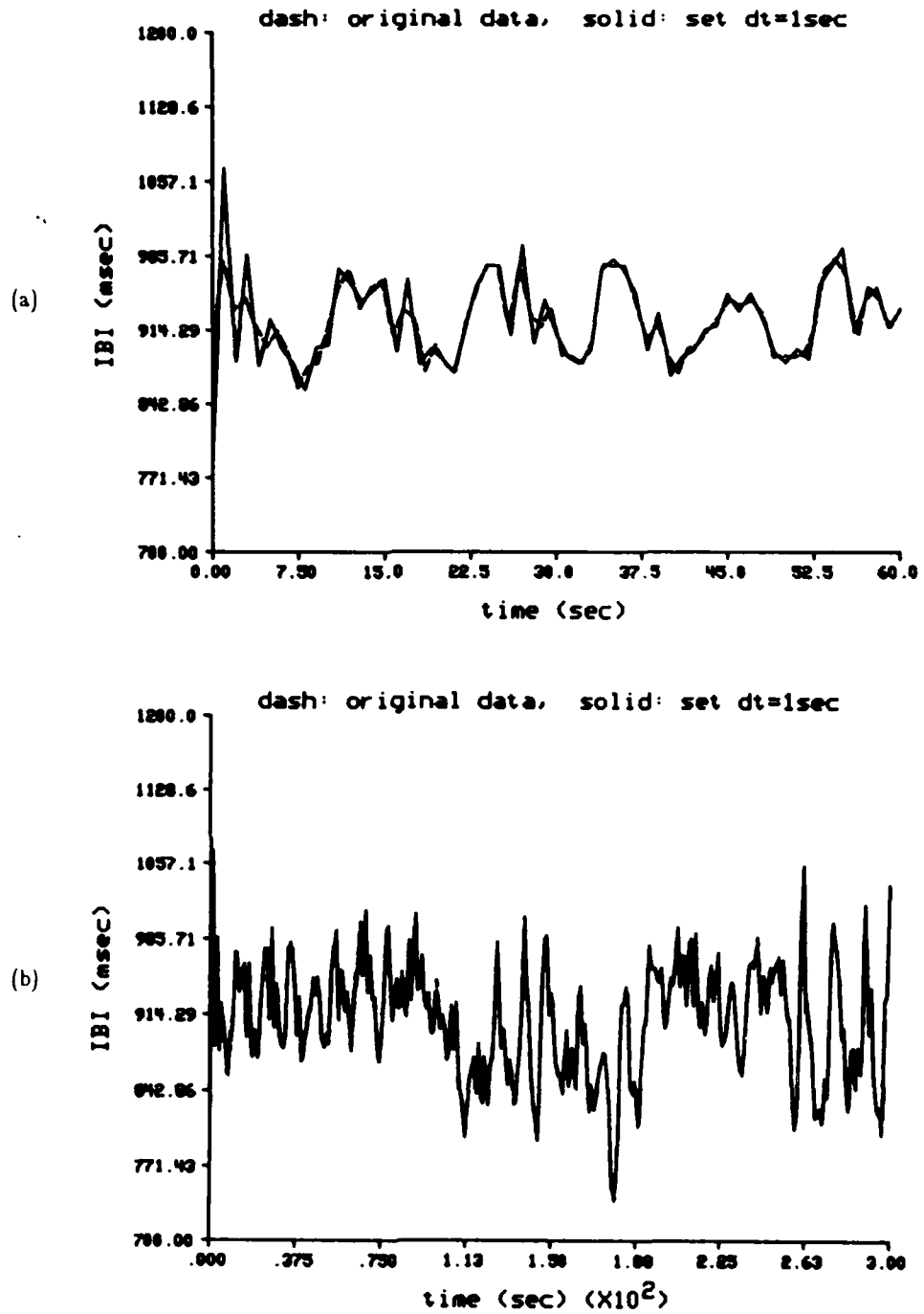


Figure 6.2.1 The Low Pass Filtered Event Series (LPFES with $dt=1\text{sec}$)
 Superimposed on the Original IBI Data for
 Subject 6 - 2 Choice - 250 ms IBI

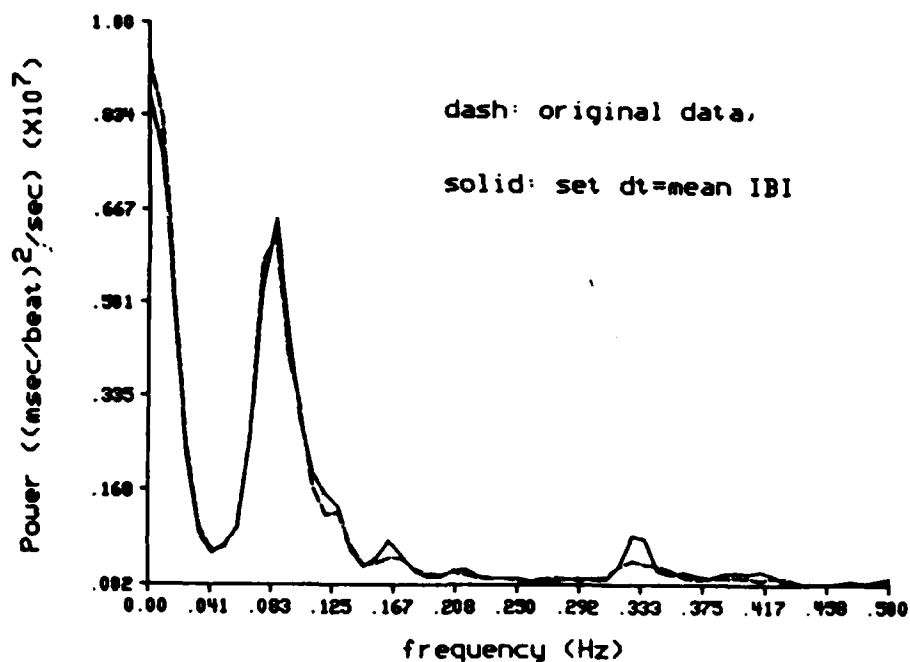


Figure 6.2.2 PSE for Original IBI vs PSE for LPFES ($dt = \text{mean IBI} \approx 906 \text{ ms}$)
Subject 6 - 2 Choice - 240 msec

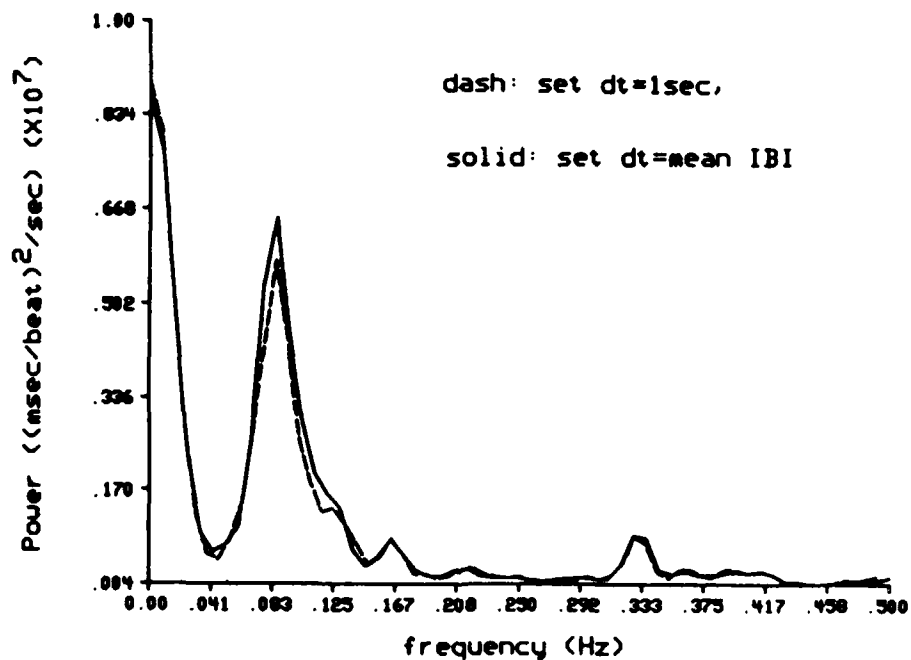


Figure 6.2.3 PSE for Original IBI vs PSE for LPFES ($dt = 1 \text{ msec}$)
Subject 6 - 2 Choice - 240 msec

would also place the maximum frequency at 0.55 Hz. The frequency axis for the spectrum of counts method (LPFES) is a true linear frequency scale. The frequency axis for the spectrum of intervals method is a linear approximation with the degree of linearity increasing as one gets closer to the origin and the coefficient of variation decreases.

Note the emergence of a sharp peak in the PSE from the LPFES at 0.167 Hz and 0.333 Hz. While these peaks are also present in the spectrum of counts, they are not as pronounced. A peak was consistently located at 0.167 Hz in the PSE for all subject and loading conditions, though its amplitude varied greatly according to subject. It is believed that this peak is due to entrainment of the heart rate by the warning tone which was presented every six seconds to signify the start of a new trial. Note that the stimuli were not periodic since their location was varied relative to the warning tone. Mulder et al. (1973) also reports finding a peak at the stimulus frequency in the power spectrum. The peak at 0.333 Hz is believed to be a harmonic of the peak at 0.167 Hz though it could not always be identified due to its overlap with the respiratory band in the power spectrum. The mean inter respiratory rate for this subject and loading condition was 2808 ms with a standard deviation of 394 ms. This would center the respiratory peak at 0.356 Hz. The degree of sinus arrhythmia, as indicated by the percent of power in the respiratory band of the PSE, varied greatly from subject to subject. The current subject, Subject 6, showed one of the lowest degrees of sinus arrhythmia.

The superiority of the spectrum of counts (LPFES) PSE over the spectrum of intervals (original IBI) PSE in identifying time synchronized (as opposed to beat synchronized) events is the primary reason that it was used in the analysis. One possible indicator of loading that was considered was the relationship between these two PSEs. If the time increment of the LPFES is set to the mean IBI, the points on the PSE frequency axis should roughly align with those on the PSE from the original IBI data. The waveform obtained by a point by point subtraction of these two PSE did not appear to yield any measures consistent with the loading. For the remainder of the analysis of heart rate variability, the time increment for the French-Holden algorithm was set to 1 sec ($F_{max} = 0.5$ Hz). Figure 6.2.3 is an overlay of the LPFES ($dt = 906$ ms) PSE shown in Figure 6.2.2 with the PSE computed for a LPFES ($dt = 1.0$ sec) to dispel any doubts that the peaks may somehow be dependent on the low pass filter in the interpolation algorithm.

The mean of a series is usually removed before computing the power spectrum in order to remove the large DC component from the spectrum. If it is believed that the series has a trend which is not relevant to the process being measured, then the trend can also be removed before computing the power spectrum. This will remove the power at DC and the extreme low frequency end of the spectrum. Some of the data records did indicate a trend, though it did appear to vary by subject and loading condition. A linear regression was performed on all of the cases, but the slope did not appear to be consistent with the workload. Working with the differences of the points is another technique that is used when it is believed that extraneous trends are present. This is the dual of taking the derivative of a continuous waveform, which would multiply its resultant PSE by a ω^2 window. The net effect is to reduce the power in the low frequency bands while emphasizing the power in the high frequency bands.

In order to be complete, PSEs were computed using each of the above prefiltering operations: demeaning - removal of the mean, detrended - removal of the best fit line, and differenced - using the difference between successive IBI. A PSE was also computed using the heart rate, which is the output of the French-Holden algorithm, rather than the IBI. The mean rate was subtracted out before the PSE was computed. The four types of PSE were all computed using the LPFES with a time increment of 1 sec. Note that this time increment was greater than the mean IBI for Subjects 4, 5 & 6, less than the mean IBI for Subjects 8 & 9 and about the same as the mean IBI for Subject 7. Figure 6.2.4 shows four PSEs computed in the above fashion for Subject 8 under both the 4 Choice - 62.5 ms ISI (high) loading and the 2 Choice - 250 ms ISI (low) loading. The large scale on the power axis is due to working in a millisecond time scale for the inter-beat intervals. Note that when using the French-Holden algorithm it is possible to get aliasing in the PSE. The PSE from the differenced beats emphasize the aliasing due to its nonlinear weighting of the PSE. It was judged that the aliasing was not significant in the bands of interest and the aliasing for the low load condition shown in Figure 6.2.4 was typical. The degree of aliasing indicated by the high load condition in Figure 6.2.4 was fairly rare.

Note the large reduction in power in the high loading case compared to the lower loading case. Six loading conditions for six subjects and four types of prefiltering yields 144 graphs of the type shown in Figure 6.2.4. There are too many conditions to compare manually. It was decided to analyze this data by comparing the power in four frequency bands. The bands were

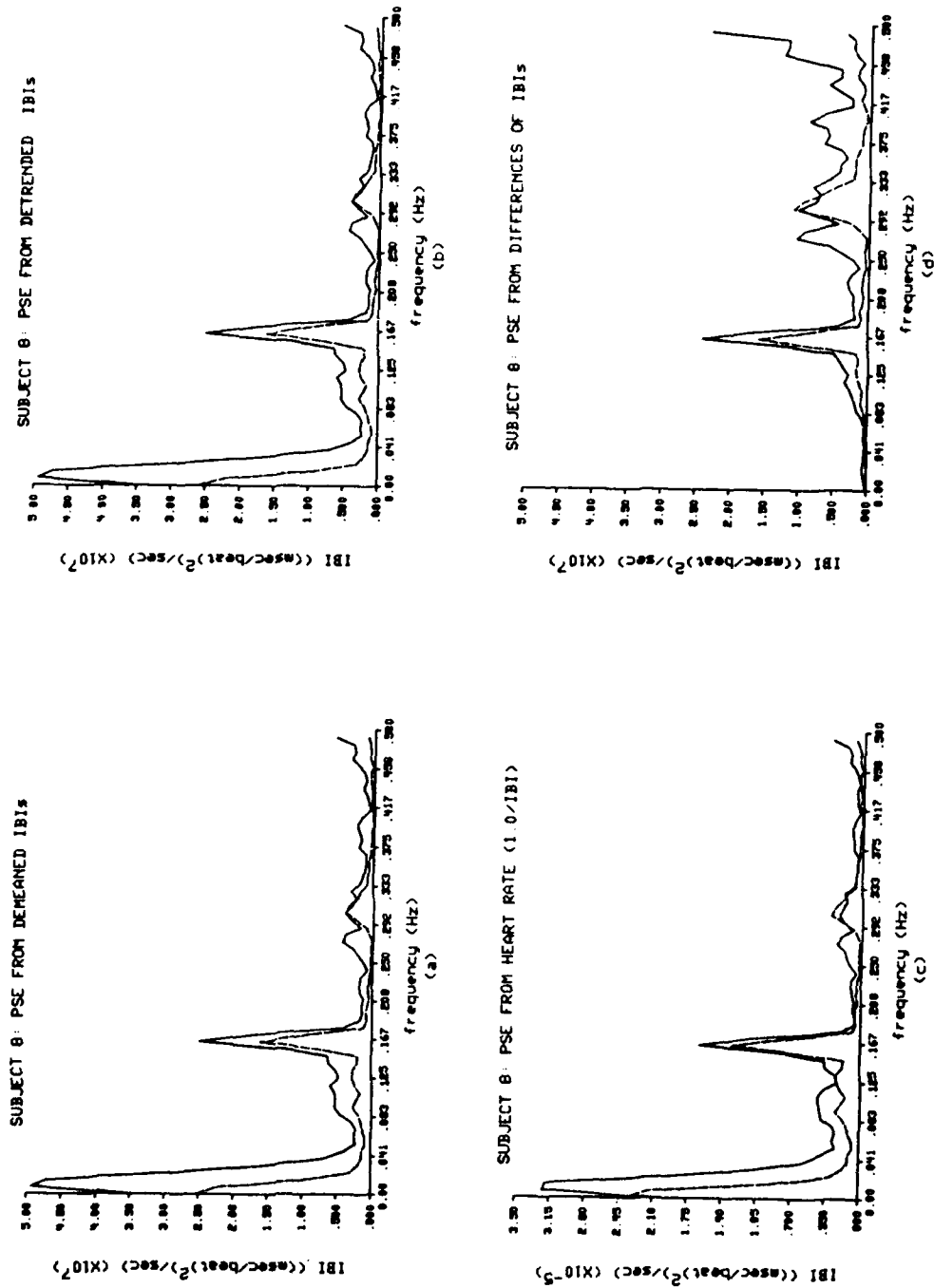


Figure 6.2.4 LPFES Spectrum for Various Prefiltering

chosen to correspond with the stimulus frequency as well as the three main physiological factors known to effect heart rate (Sayers 1973); thermal regulation, blood pressure control, and respiration.

The location of the stimulus band was easy to determine since it is centered at $1/(\text{stimulus period})$ or 0.167 Hz. It was decided to sum up the power in the band from 0.1484 to 0.1796 Hz (5 points wide) to represent the contribution of the stimulus to the PSE. It was not always clear from the PSE where the respiration peak was located so it was decided to use the respiration measure itself to define the respiration band. For the low load condition shown in Figure 6.2.4 the mean respiratory interval was 3553 ms (≈ 0.28 Hz) with a standard deviation of 537 ms. For the high load condition it was 3210 ms (≈ 0.31 Hz) with a standard deviation of 298 ms. The respiratory band was defined to start at $1/(\text{mean} + \sigma)$ and end at $1/(\text{mean} - \sigma)$. This corresponds to bands of 0.245 - 0.332 Hz and 0.285 - 0.343 Hz for the low and high loading cases respectively. Note that this aligns with the peak evident in the high load case and allows us to make an objective decision in the low load case where a single peak cannot be isolated.

The choices for the thermal and blood pressure band are not as straight forward as for the stimulus and respiration band. We do not have independent measures of body temperature or blood pressure. Sayers (1973) has correlated the heart rate with independent measures of body temperature and blood pressure. He reports the thermal band to be centered at 0.025 Hz and the blood pressure band to vary from 0.07 to 0.10 Hz (see Figure 3.1.1). The frequency increment, when using the aforementioned procedure for computing the PSE, is 0.00781 Hz. The power at a particular frequency is spread over several adjacent points in the frequency domain, commonly called leakage effect, when the PSE is computed from a finite data length. Based on the above facts and from viewing PSE across the subject and loading cases for peak locations, it was decided to define the thermal band as DC to 0.0391 Hz and the blood pressure band as 0.0547 to 0.1250 Hz. The blood pressure activity is not clearly indentifiable in Figure 6.2.4, but in other subjects this band was found to align with significant activity. The thermal band chosen was in good correlation with the activity shown in Figure 6.2.4.

The bar graph shown in Figure 6.2.5 shows the total power (DC to 0.5 Hz) for all subjects, loading and prefiltering conditions. Figures 6.2.6, 6.2.7 and 6.2.8, 6.2.9 are bar graphs which show the power in the thermal, blood

pressure, stimulus, and respiration bands respectively. The axis for the power (ordinate) in Figures 6.2.5 to 6.2.9 is an offset dB scale that ranges from 0 to 30 dB. The offsets used between bar graphs were not constant in order to be able to show the bars at a reasonable height. The offset within an individual bar graph was constant and is indicated on the axis.

Before discussing these graphs which summarize the spectral nature of the heart beat, the behavioral measures of response (Table 5.2.2) should be reviewed. The most significant changes in behavioral performance occurred for three of the binary comparisons between loading (columns 3, 5 & 6 in Table 5.2.2). The three binary comparisons were:

- 4 Choice - 62.5 ms vs. 4 Choice - 250 ms(ISI Loading)
- 4 Choice - 62.5 ms vs. 2 Choice - 62.5 ms(Choice Loading)
- 4 Choice - 62.5 ms vs. 2 Choice - 250 ms(High-Low Loading)

Note that all of the comparisons involved the highest loading level of 4 Choice - 62.5 ms. It may be that comparisons between the other levels of loading are not as significant because the subject was not sufficiently loaded to produce a marked behavioral or physiological difference. We will refer to these three comparisons of loading by the above names which are given in parentheses. In looking at the bar graphs (Figures 6.2.5 to 6.2.9) this corresponds to the sixth bar in each group being lower than the third, fourth, and fifth bar if the power in each band decreased with loading. ISI loading is indicated by comparing the fourth bar to the sixth bar. Choice loading is indicated by comparing the fifth bar to the sixth bar. High-low loading is indicated by comparing the third bar to the sixth bar.

In examining total power shown in Figure 6.2.5, the demeaned and detrended method yield the correct relationships for all three comparisons for all subjects with the exception of Subject 6. The heart rate yields similar results except that it also fails for the high-low comparison for Subject 5. The differences method fails all three comparisons for three of the subjects and passes all three comparisons for the other three subjects. This can be attributed to its extremely nonlinear windowing of the PSE. Note that all four prefiltering methods passed all three comparisons for the mean of the six subjects. Extreme caution should be exercised when simply working with the mean since the mean of the difference method passed all three tests while half of the individual subjects failed.

In comparing the four filtering methods when looking at the individual frequency bands, all methods gave roughly similar performance. The

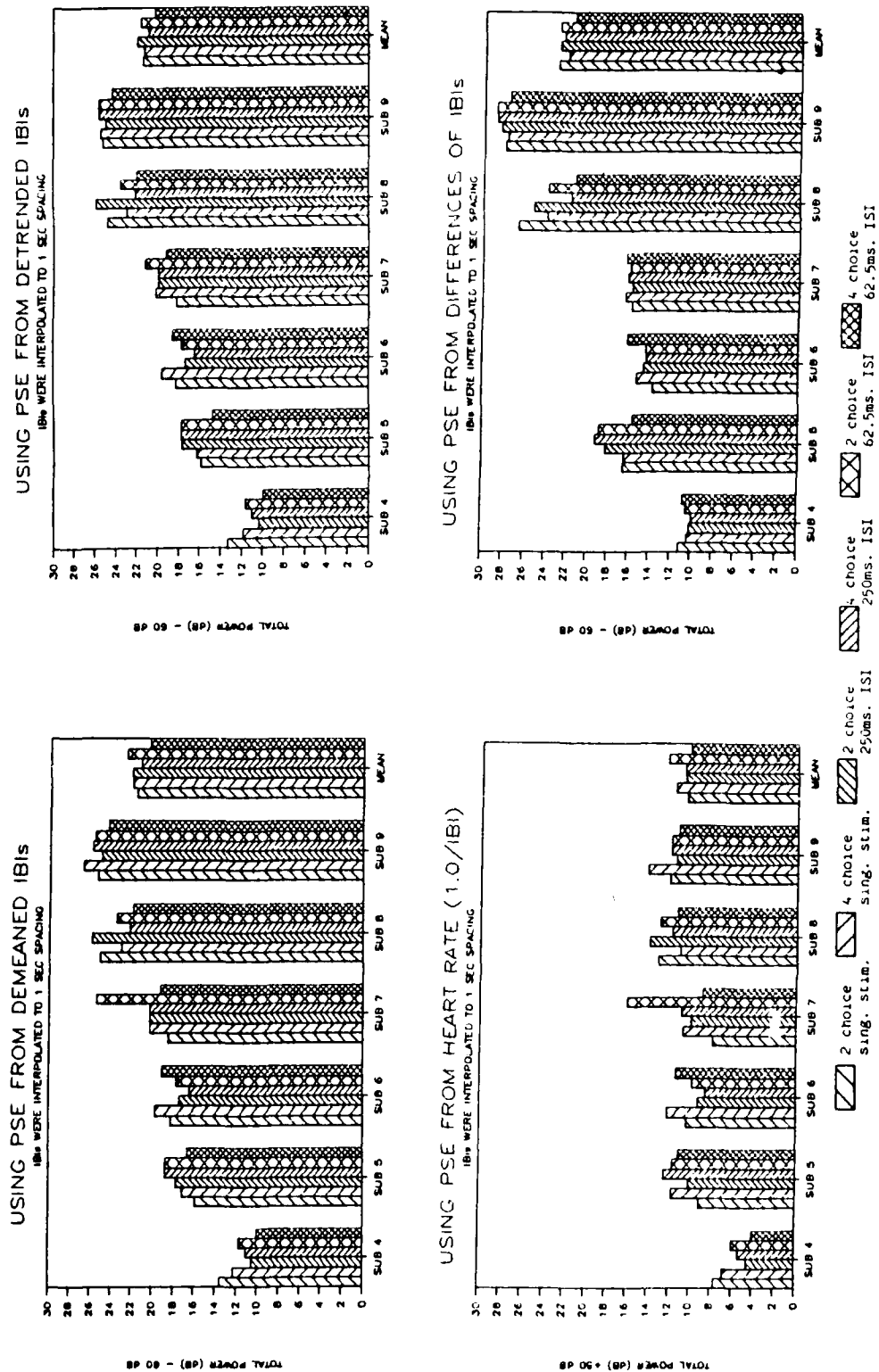


Figure 6.2.5 Total Power (DC to 0.5 Hz) in the LPFES Spectrum

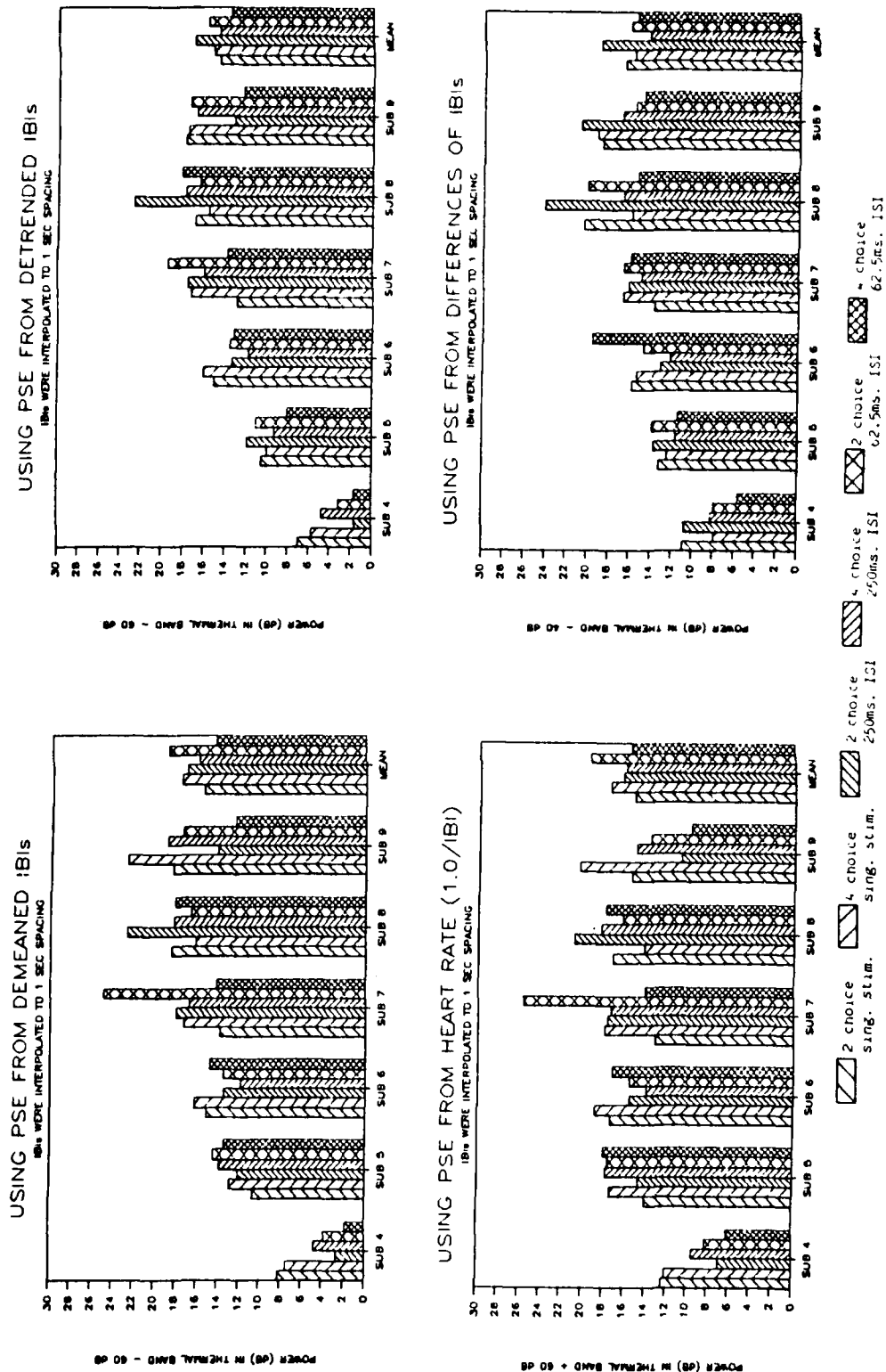


Figure 6.2.6 Power in the Thermal Band (DC to 0.0391 Hz) of the LPFES Spectrum

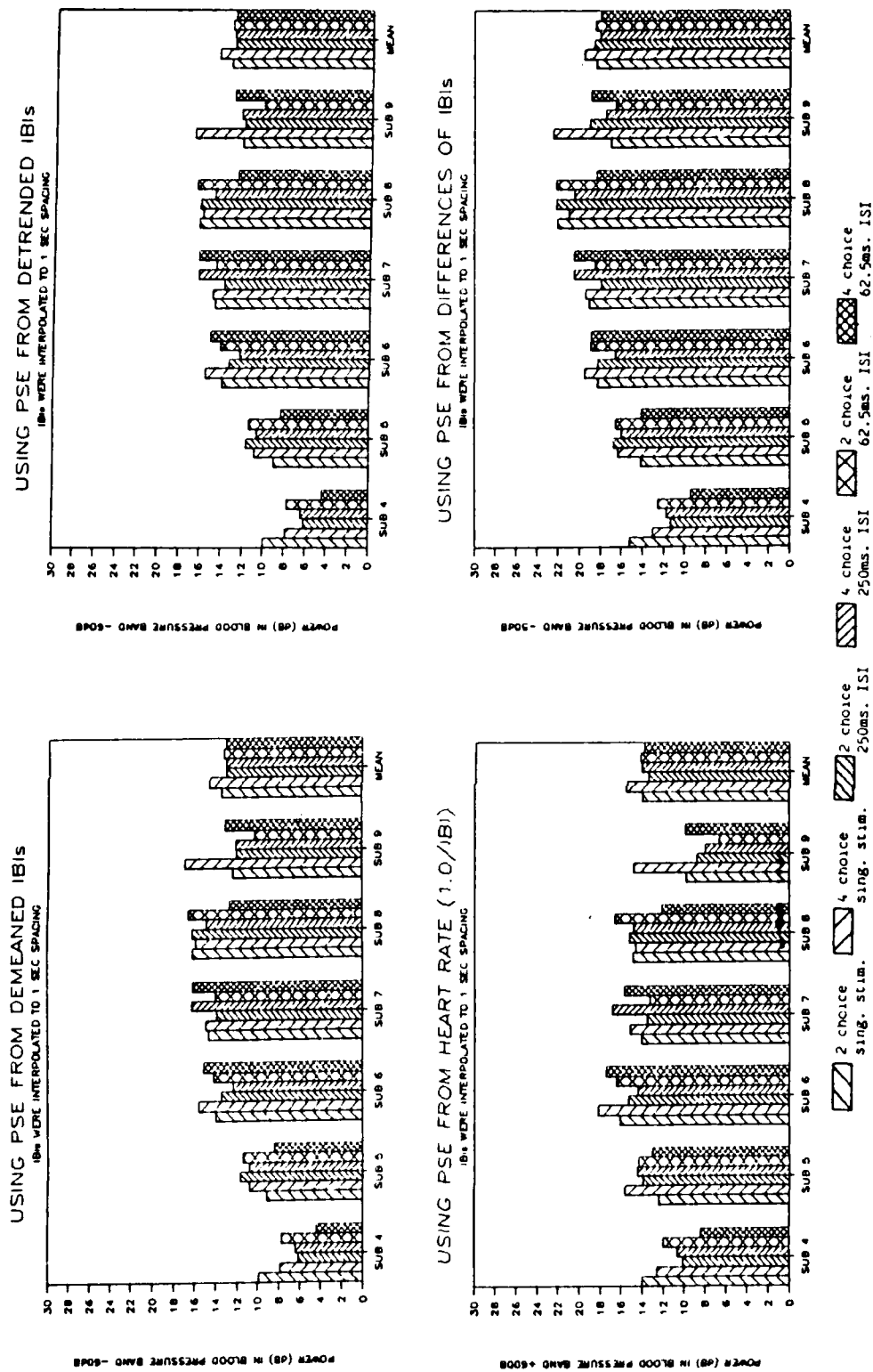


Figure 6.2.7 Power in the Blood Pressure Band (0.0547 to .1250 Hz) of the LPFES Spectrum

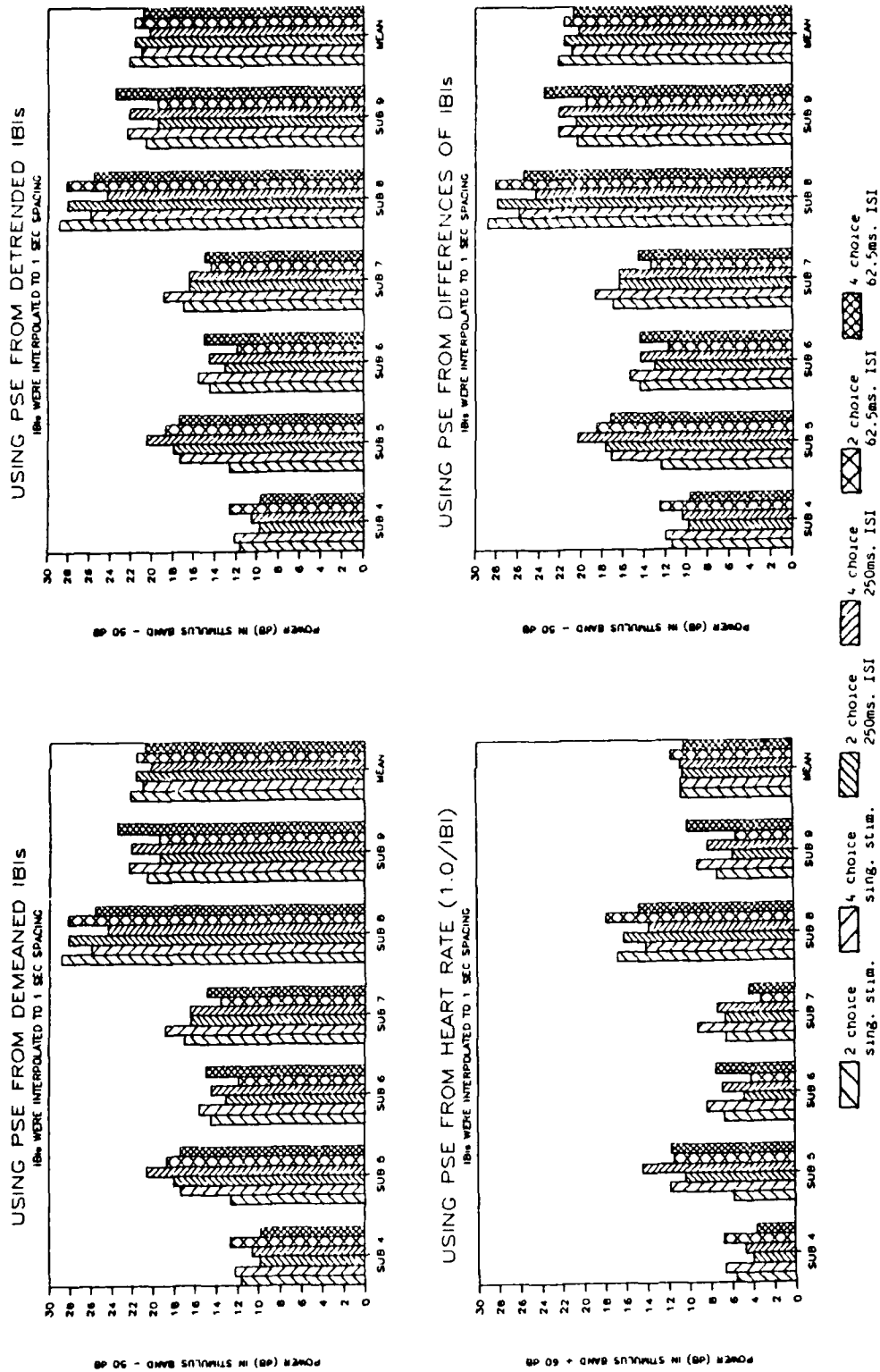


Figure 6.2.8 Power in the Stimulus Band (0.1484 to 0.1796 Hz) of the LPFES Spectrum

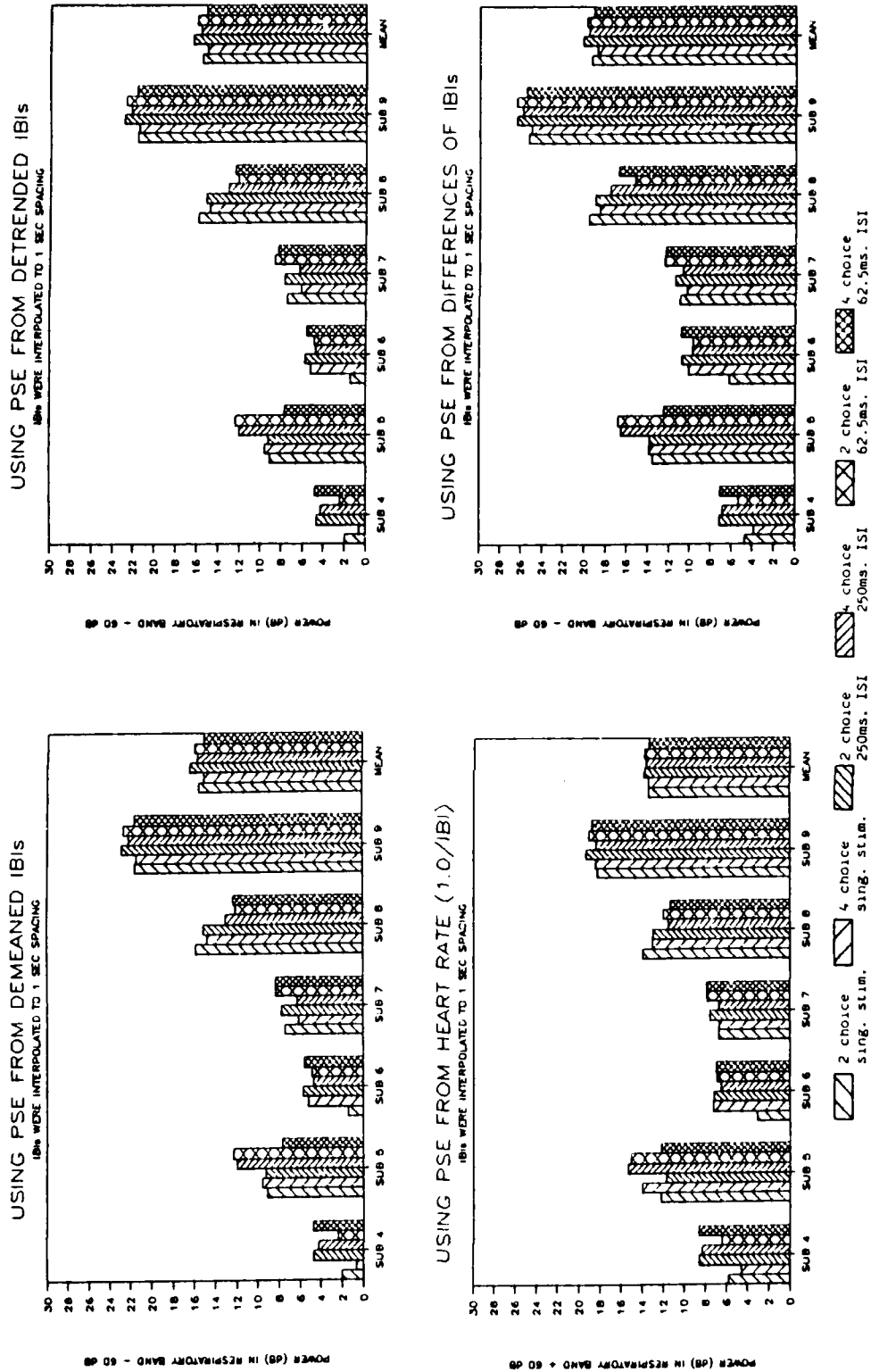


Figure 6.2.9 Power in the Respiration Band of the LPFES Spectrum

problem with the nonlinear weighting of the frequency axis by the differences method is not very severe when the power is computed from only a few adjacent points. There was no single band for any of the prefiltering methods that could correctly classify any of the loading comparisons for all six subjects. If the acceptance criterion is dropped to 5 out of 6 subjects, then the thermal band is the only band that can correctly classify some of the loading comparisons. The demeaned thermal band can distinguish the choice comparison, while both the detrended and differenced thermal band can distinguish the high-low and choice comparison.

The relative heights of the bars in the graph of the PSE for the demeaned prefiltering condition, shown in Figure 6.2.5, should correspond to the relative heights of the bars of the graph of the standard deviation, shown in Figure 5.3.2. The only notable exception to this is Subject 7 under the 4 choice - 62.5 msec loading condition. The IBI plotted in Figure 5.3.5 is from this subject and loading case. The initial few seconds at the start of the block had a IBI which was significantly lower than mean (by ≈ 300 ms). In computing the PSE, these points were truncated. They were not truncated in the computation of the standard deviation which explains the inflated estimate. If this error is compensated for, then the standard deviation will have same classification performance as the total power. The standard deviation estimate computed by the mean square successive difference (MSSD), as shown in Figure 5.3.3, should not be subject to this bias. The MSSD estimate of the standard deviation could only classify four out of the six subjects for the choice comparison. Its performance for the other comparisons was lower so that its use as a classifier is dubious at best.

A relative measure of the power in a band may indicate effects even though the absolute measure of the power in the band might not. Figure 6.2.10 is a bar graph which indicates the percent of the total power lying in each of the four bands using the demeaned method of prefiltering. Six binary comparisons can be made when using four distinct bands. Figure 6.2.11 shows the relative power between the bands for the six binary comparisons. Figure 6.2.11 uses a dB scale so that a negative number indicates that the first band had less power than the second band. Unfortunately, none of the graphs in Figures 6.2.10 & 6.2.11 can distinguish between any of the loading comparisons for all six subjects. At the five out of six subject acceptance level, there is a slightly greater percentage of power in the stimulus band for the high-low comparison except for Subject 7. This also shows up as a decrease in the ratio of the thermal band power to the

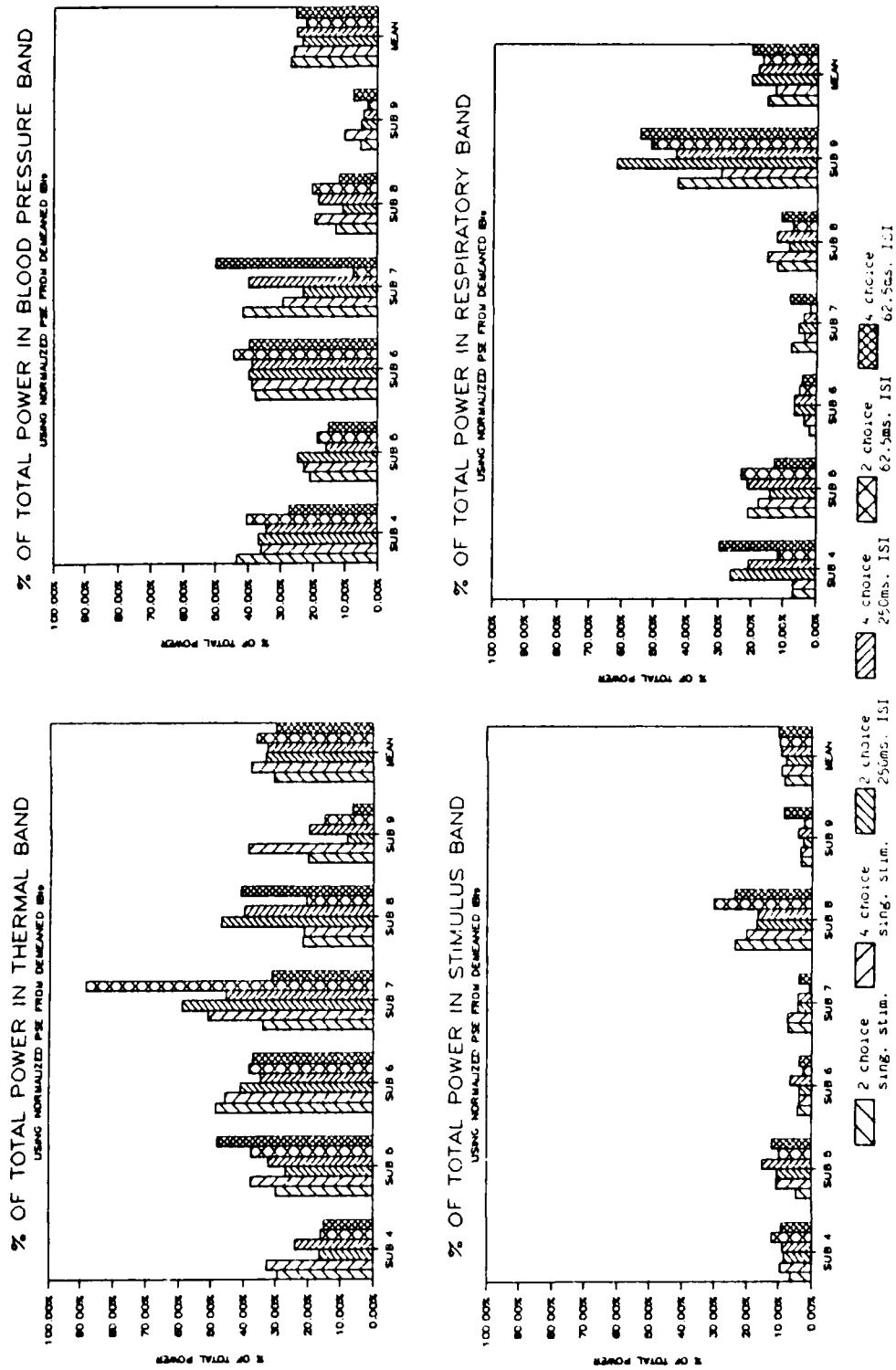


Figure 6.2.10 Percent Power in the Bands of the LPFES Spectrum

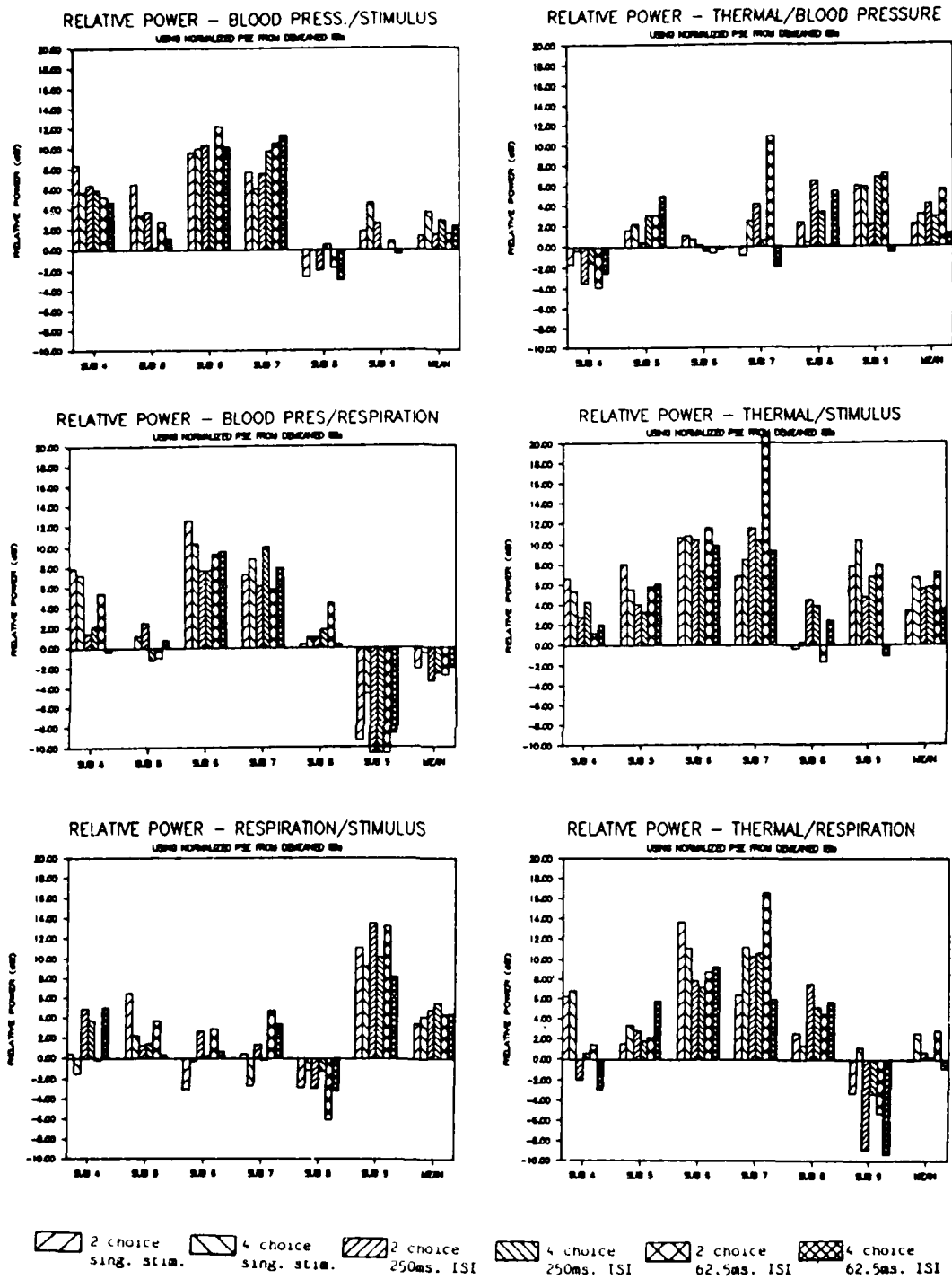


Figure 6.2.11 Relative Power in the LPFES Spectrum

stimulus band power for the high-low comparison except for Subject 5. The ratio of blood pressure power to stimulus band power also decreases for the choice comparison except for Subject 7. The best classifier appears to be the total power in the demeaned or detrended PSE of the LPFES.

6.3. P300 as a Measure of Loading

The P300 or positive peak occurring in the event related potential (ERP) waveforms at ≈ 300 ms after stimulation has been extensively studied as an indicator of cognitive activity. Unfortunately, the procedures to identify and measure P300 have been far from consistent and even vary within the same research group. The use of P300 as a measure involves three steps; prefiltering, identification of the peak and computation of the measure from the identified peak.

The ERP is usually low pass filtered to reduce any alpha activity that might hinder the identification of P300. Magliero and Donchin (1984) used 3.5 Hz (-3 dB) low pass prefiltering, while Kutas and Donchin et al. (1977) used 6.29 Hz low pass prefiltering. As part of the signal processing for this research, a sharp cutoff low pass filter was designed. The Remez Exchange Algorithm (McClellan et al. 1979) provides the design of a low pass filter by specifying the passband and stopband. This is a safer approach than simply specifying the 3 dB point of a filter since it eliminates the possibility that the transition region of the filter may extend into the desired pass and stopbands. It was decided that the prefiltering should consist of a low pass filter with a passband ending at 3.5 Hz and a stopband starting at 8 Hz to block the alpha activity. The resultant filter provided 20 dB of attenuation in the stop band with a 0.15 dB ripple in the passband and a -3 dB point at 5.6 Hz. The result of this filtering on the average response (P3 electrode) for several subjects is shown in Figures 6.3.1, 6.3.2 and 6.3.3 for the single stimulation, 250 ms ISI, and 62.5 ms ISI loading conditions, respectively. Both 2 choice and 4 choice conditions are shown in the figures.

Peak identification has traditionally consisted of finding the largest amplitude within a specified interval. The interval used, however, varies with the researcher. Kavis and Donchin (1983) used a 300 to 750 ms interval, while McCarthy and Donchin (1981) used a 200 to 1500 ms interval. Virtually every paper that deals with P300 uses a different interval. The current research allows the recorded data to determine the interval based on

ELECTRODE P3
3.5 Hz LPF

60 msec ISI

2 CHOICE CASE = solid
4 CHOICE CASE = dash

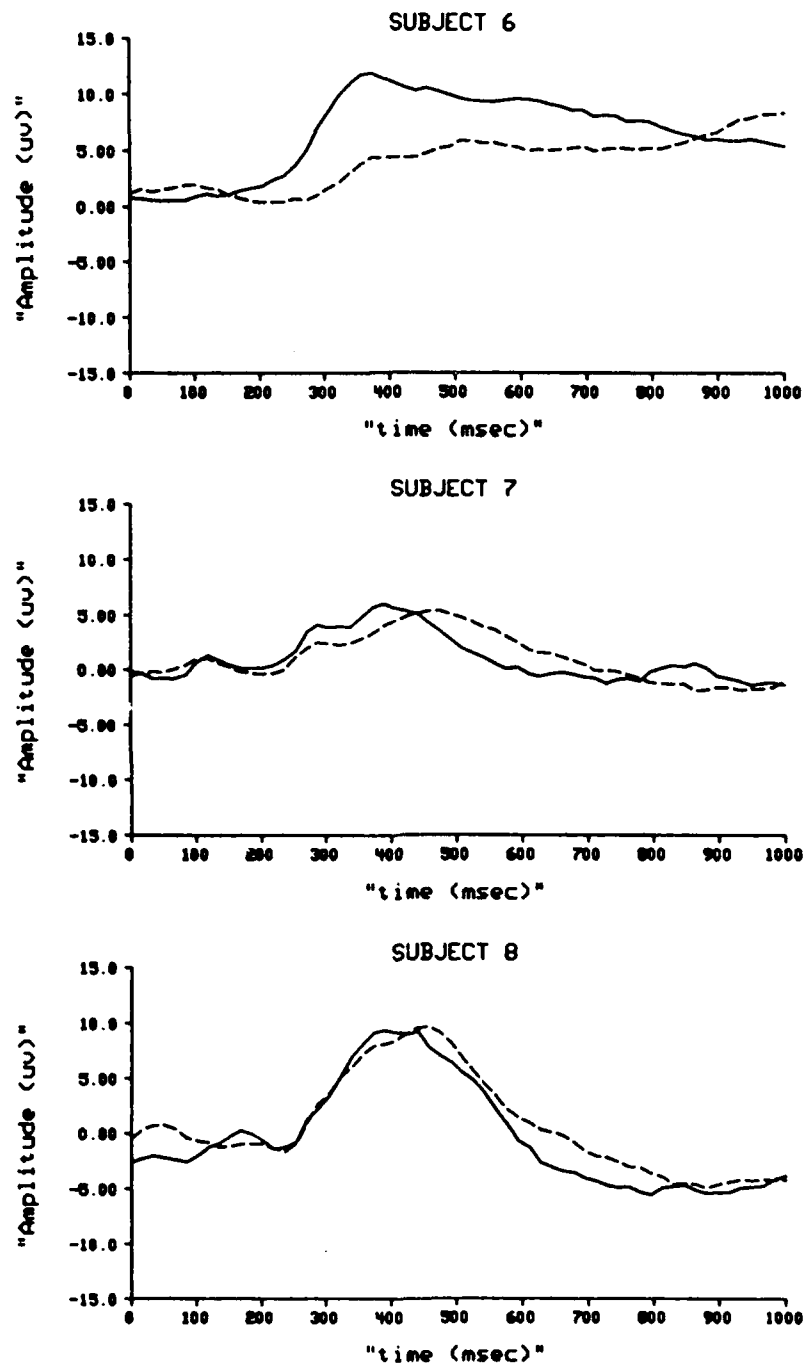


Figure 6.3.1 3.5 Hz LPF Averages: P3-Electrode, Single Stimulation

AD-A188 285

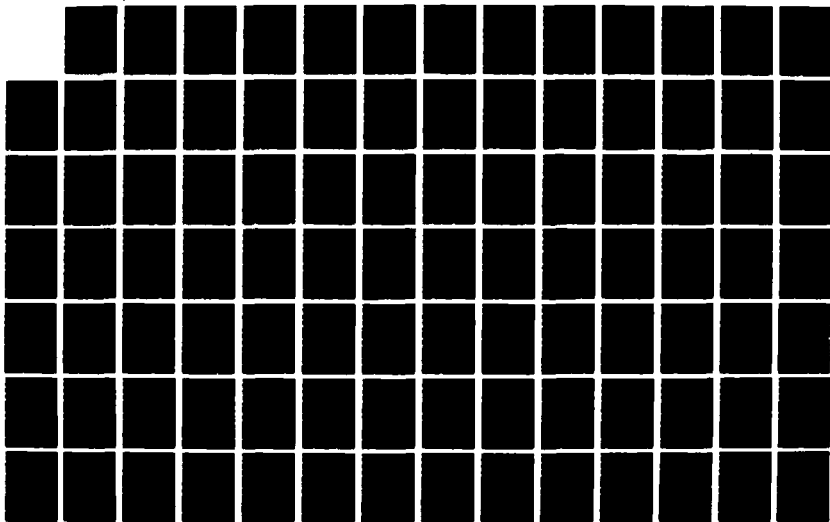
ELECTROMAGNETIC METRICS OF MENTAL WORKLOAD(U) PURDUE
UNIV LAFAYETTE IN EEG SIGNAL PROCESSING LAB
J I AUNON ET AL SEP 87 AFOSR-TR-87-1663 AFOSR-85-0313

2/3

UNCLASSIFIED

F/G 5/8

NL





ELECTRODE P3
3.5 Hz LPF

250 msec ISI

2 CHOICE CASE = solid
4 CHOICE CASE = dash

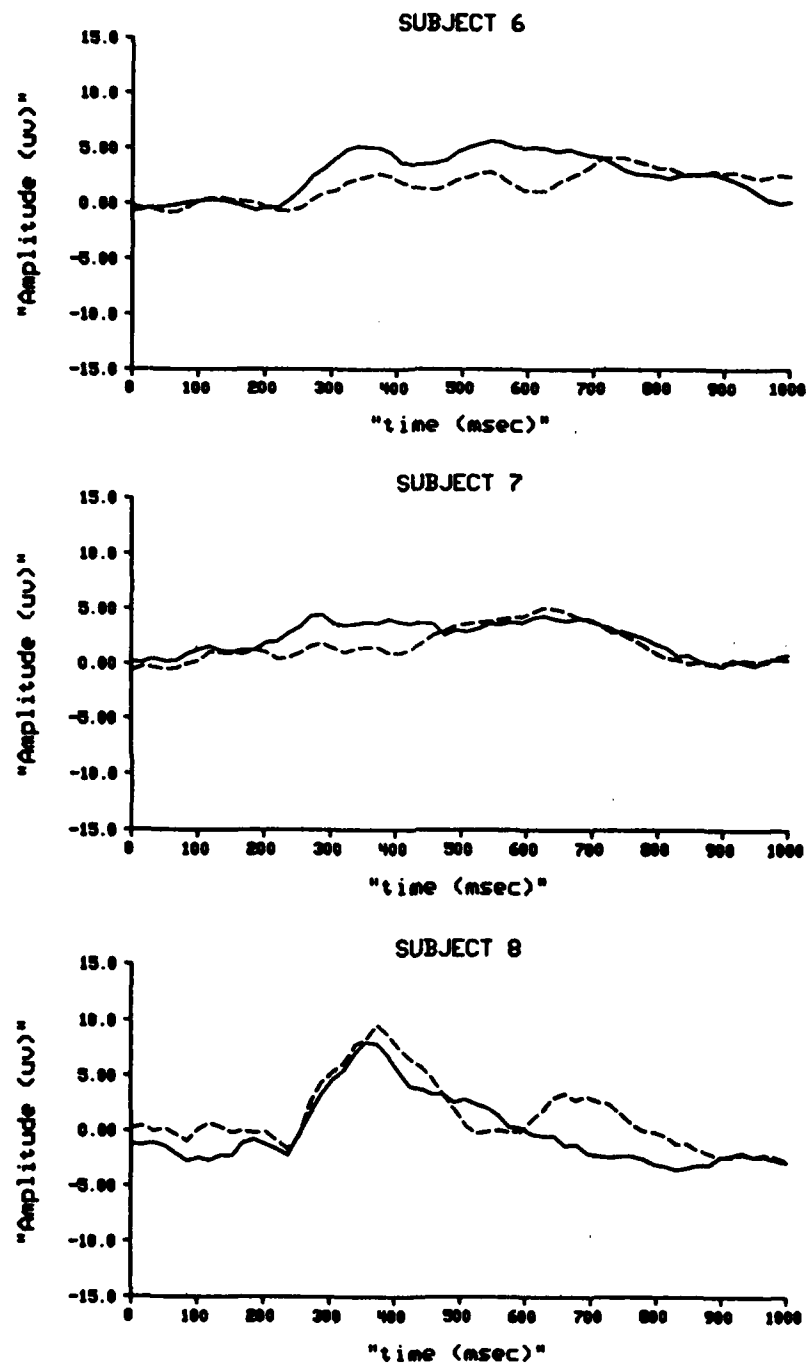


Figure 6.3.2 3.5 Hz LPF Averages: P3-Electrode, 250 msec ISI

ELECTRODE P3
3.5 Hz LPF

SINGLE STIM

2 CHOICE CASE = solid
4 CHOICE CASE = dash

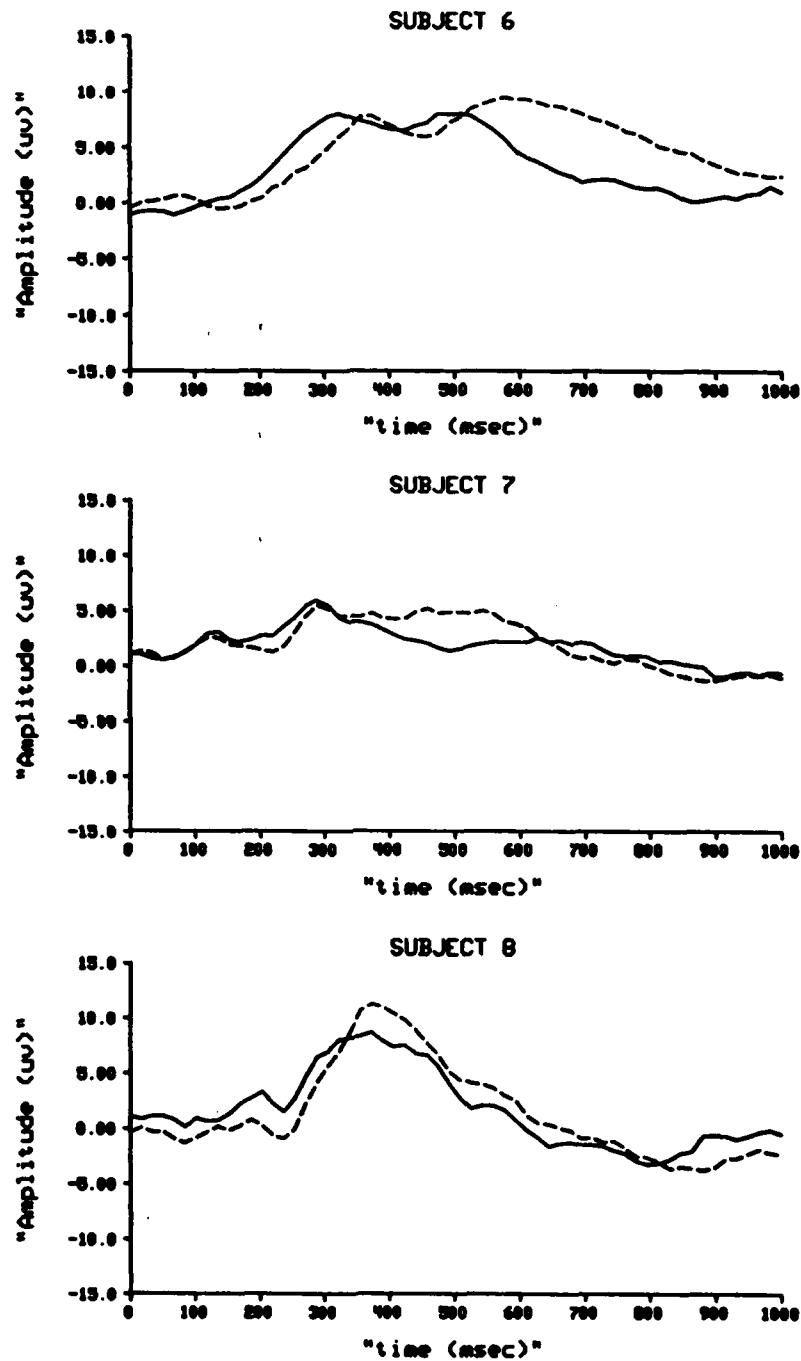


Figure 6.3.3 3.5 Hz LPF Averages: P3-Electrode, 62.5 msec ISI

a statistical test rather than experimenter bias. The mechanisms for this procedure is described in the subsequent section entitled "Latency Corrected Average".

The purpose in locating the peaks is so that some metric can be calculated which may prove useful in determining levels of cognitive activity. Some researchers avoid the issue of trying to locate a specific peak in each trial by using a discriminant function (Squires & Donchin, 1976; Duncan-Johnson & Donchin, 1981). This discriminant function uses a sum of weighted values applied to the amplitude at several latencies and across several electrodes. These weightings were determined by comparing classification accuracies using a Step-wise Discriminant Analysis (SWDA) procedure for cognitive tasks across a large subject base. Other measures that have been computed include the peak latency and amplitude or area computations. Latency is usually computed from the peak in the average (McCarthy & Donchin, 1981; Magliero & Donchin 1984) and has been reported to vary from 210 ms to 550 ms after stimulation (John, 1977). Amplitude measures can be taken from the amplitude of the largest peak in the conventional average, the amplitude of the largest peak in the Woody Correlation Average (Kutas & Donchin et al., 1977) or the average from aligning all of the largest peaks found in the individual trials (Duncan-Johnson & Donchin, 1981). The area of an interval around 300 ms has also been used as a measure since it minimizes the effect of latency jitter on the peaks (Israel et al., 1980). Israel also computed the latency jitter as a measure derived from P300. The present research uses the latency corrected average to locate and compute measures for P300.

6.4. The Latency Corrected Average

The (LCA) latency corrected average (McGillem & Aunon, 1977; Aunon & McGillem, 1979; Begleiter et al., 1983) was chosen as the method of identifying cognitive measures from the ERP. For the purposes of discussing the LCA, the term "LCA peak" shall refer to the average peak computed by the program and the term "trial peak" shall refer to a peak from an individual trial that was used in computing the average peak.

The LCA procedure was selected for several reasons. Trial peaks are selected only from statistically significant regions which are determined individually for each block (50 trials). An LCA peak is computed for each valid region, rather than just using a single region to represent the entire response. The LCA allows for the possibility of a trial to have a null

response; i.e. it is not required that every trial have a valid peak in each region. Trial peak statistics (percent occurrence, mean latency, latency variance and mean amplitude) are computed only from valid peaks within the region. The latency of an LCA peak is determined by averaging the trial peak latencies, rather than averaging the peaks themselves, and then searching for the peak in the average.

The LCA procedure is implemented in the "MLCAY" program and starts by computing and applying a minimum mean square error filter (MMSE) filter to the normalized and detrended data. All peaks within the individual trials are detected by crosscorrelating the trials with a template having the generalized shape of a peak in the ERP. Histograms of both the positive and negative peaks are computed at each latency from the ensemble of trials. A sliding window is then applied to the histograms and a non-parametric sign test (Sachs sec 4.2.4, 1984) is performed to determine if the data in the window is from a positive, negative or zero median process at a specified confidence level (95%). If the sign test indicates a positive median for that window then the region is considered to be one of significant positive peaks. The record is then divided up into regions of positive, negative and no peaks. For each positive region, the largest positive peaks from each trial are aligned at the mean trial peak latency and averaged over an eleven sample point width. Negative peaks are processed in a similar fashion.

The output of the "MLCAY" program is a listing of the locations of the positive and negative peaks (up to a maximum of 10 each) and the average (11 sample points wide) of the trial peaks used to compute that LCA peak. Statistics for the LCA peak are also computed (percent occurrence of the trial peaks in the region, mean amplitude, latency mean and variance). A listing of the location and amplitude of the trial peaks used for each LCA peak is also output. Intermediate results, such as histograms, MMSE filter, etc may also be output to check program operation.

Figure 6.4.1 shows a plot of the LCA peaks computed from the same data shown as an average in Figure 6.3.3. Note that the LCA identifies some peaks that do not show up in the average due to smearing caused by latency jitter. The P300 peak has been clearly identified for all three subjects in the 400 to 500 ms range. The 4 choice - 62.5 ms ISI P300 response clearly occurs later than the 2 choice - 62.5 ms ISI P300 which is in agreement with previous findings that P300 latency increases with loading (Review in Pritchard, 1981). The negative peak in the 100-200 msec range (commonly called N200) also appeared to be slightly delayed in latency for

the higher loading. The comparisons indicated on this graph are for the choice loading comparison that was discussed in section 6.2.

Figures 6.4.2, 6.4.3 & 6.4.5 show the results of the LCA for the P4 electrode for Subjects 6, 7, & 8 respectively. The highest loading case in these figures are the dashed line in the bottom graph. The choice comparison compares the dashed line in the bottom graph to the solid line in the bottom graph. The ISI comparison compares the dashed line in the bottom graph to the dashed line in the middle graph. The high-low comparison compares the dashed line in the bottom graph to the solid line in the middle graph.

The shift in the P300 for the choice comparison, as shown in Figure 6.4.1 for the P3 electrode, is also evident for all of the subjects for the P4 electrode. It is hard to judge the ISI and high-low comparisons because the output of the LCA for the 250 ms ISI is somewhat erratic. It does not appear to find the N200 or the P300 for Subject 7 under the 4 Choice - 250 ms ISI loading condition as shown in Figure 6.4.3. For Subject 6, the LCA appears to find a N200 and P300 for each of the stimuli that were presented 250 ms apart (Figure 6.4.2 middle graph). It may be that in some of the subjects the stimuli presented 250 ms apart are be fused and treated as a single stimulation. The P300 peak does appear to shift for both the ISI and high-low loading comparisons for Subjects 7 & 8. If we only consider the first "P300" peak of Subject 6, then these comparisons also hold true.

Figure 6.4.5 shows a scatter plot of reaction time to the first stimulus (RT1) and the P300 latency. Reaction time has been shown to correlate with P300 latency under certain conditions (McCarthy & Donchin, 1981) Kutas and Donchin (et al., 1977) have shown that there is a higher degree of correlation in tasks that stress accuracy considerations over speed considerations. In the present research, the subject was instructed that speed was more important than accuracy which may account for the low correlation between RT1 and P300 latency.

6.5. Conclusions

The task description based on the hybrid capacity model was initially designed to produce 6 distinct levels of loading. The behavioral variables were not able to distinguish between the 6 distinct levels. This means that we did not have a converging criterion to justify that we had indeed changed the loading significantly in all 6 cases. The heart rate information showed a decrease in the IBI variability when comparing the highest level of

ELECTRODE P3
WIND=7, NFC=15

LOW FREQ. PEAKS 2 CHOICE CASE = solid
60 msec ISI 4 CHOICE CASE = dash

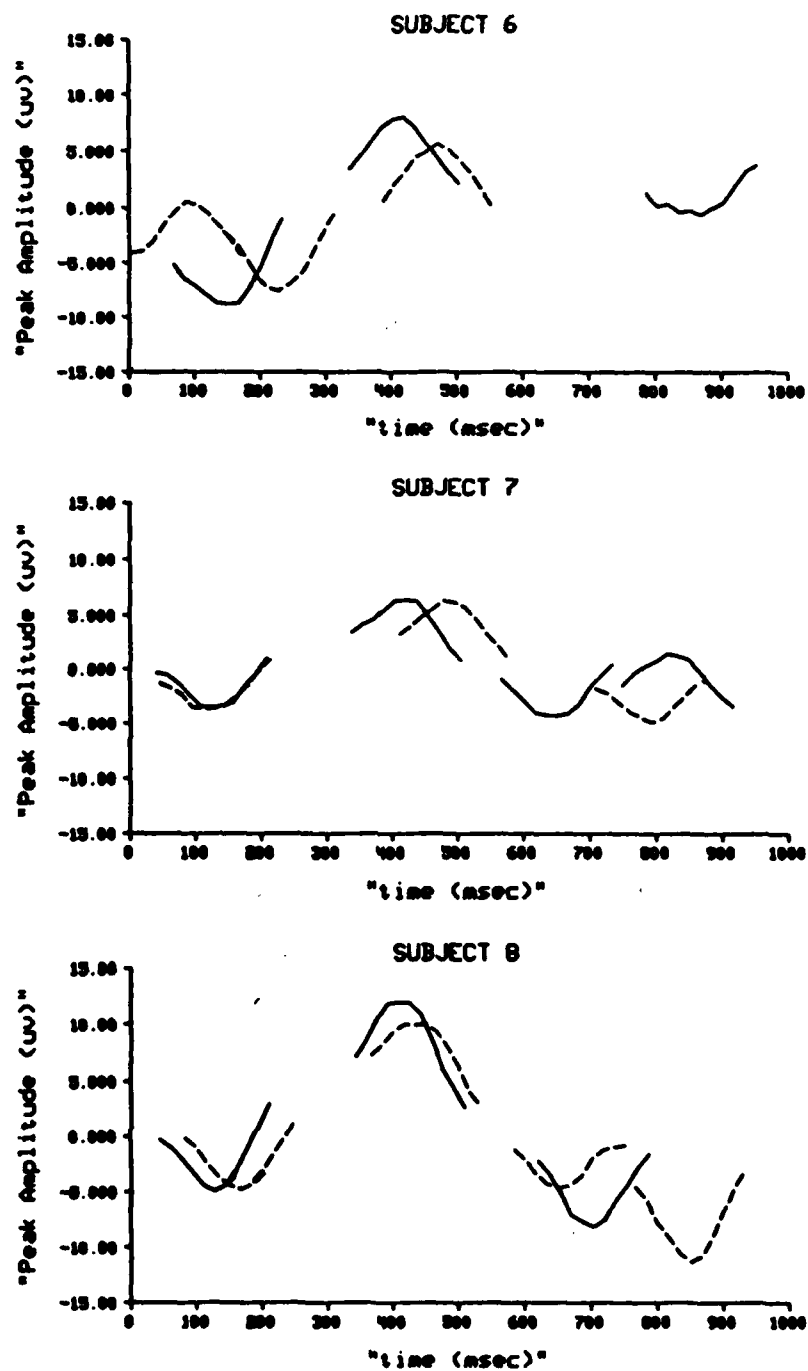


Figure 6.4.1 LCA Peaks: P3-Electrode, 62.5 msec ISI

ELECTRODE P4
SUBJECT 6

LOW FREQ.
LCA PEAKS

2 CHOICE CASE = solid
4 CHOICE CASE = dash

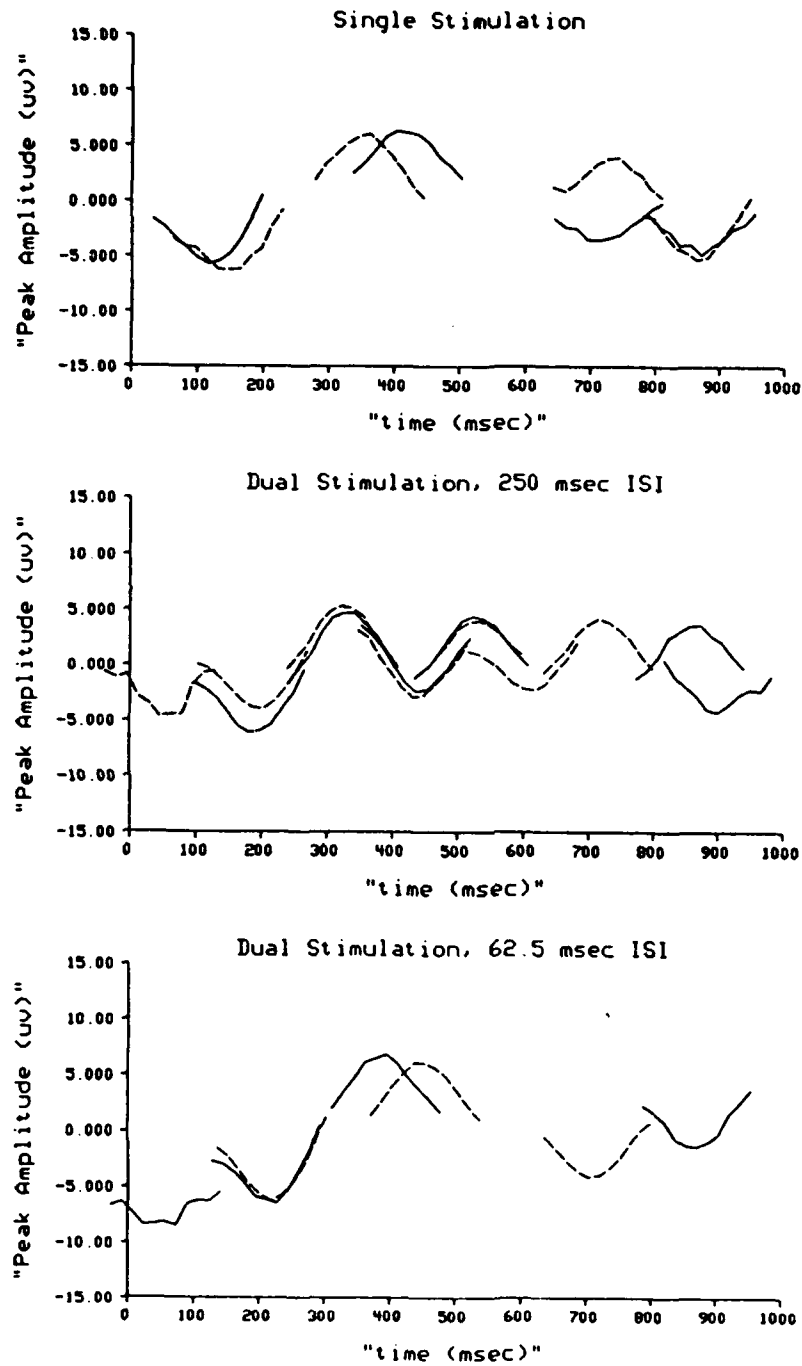


Figure 6.4.2 LCA Peaks: P4-Electrode, Subject 6

ELECTRODE P4
SUBJECT 7

LOW FREQ.
LCA PEAKS

2 CHOICE CASE = solid
4 CHOICE CASE = dash

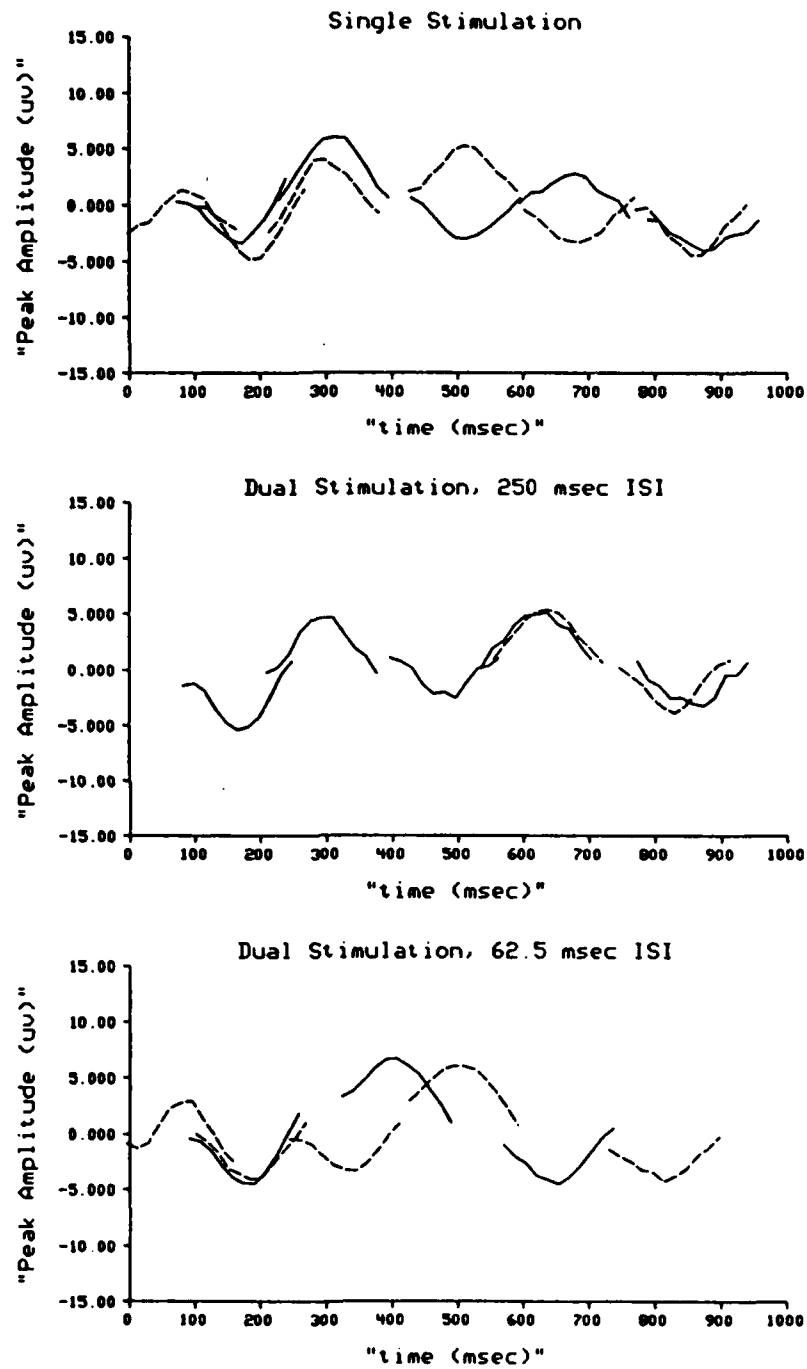


Figure 6.4.3 LCA Peaks: P4-Electrode, Subject 7

ELECTRODE P4
SUBJECT 8

LOW FREQ.
LCA PEAKS

2 CHOICE CASE = solid
4 CHOICE CASE = dash

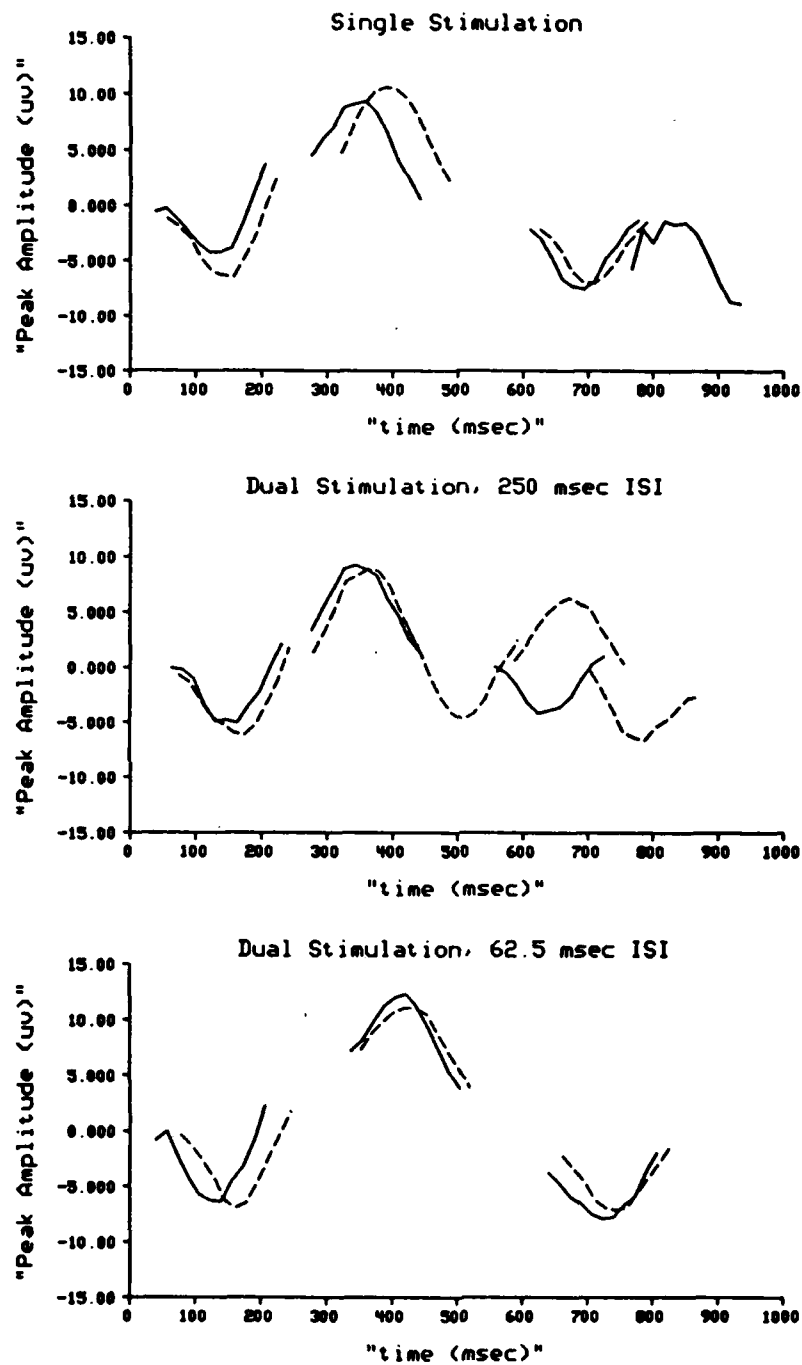


Figure 6.4.4 LCA Peaks: P4-Electrode, Subject 8

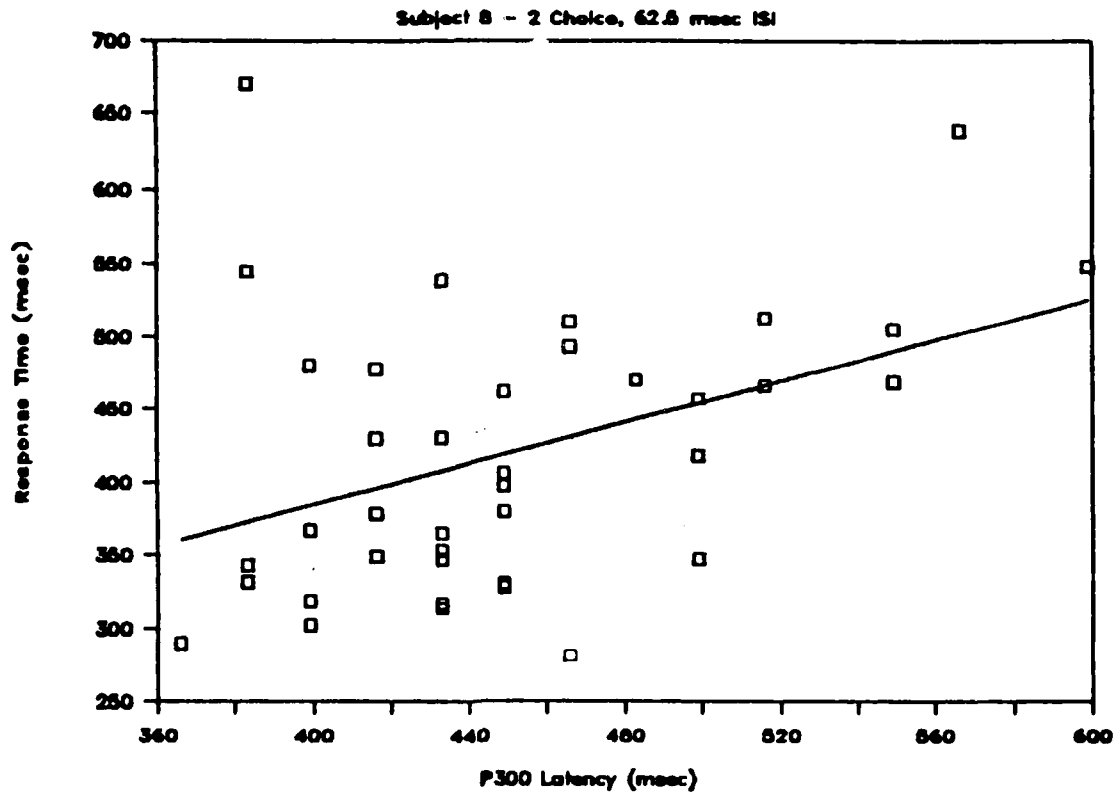


Figure 6.4.5 Response Time 1 vs P300 Latency

loading with the three other levels that were found to have a significant difference as measured by the behavioral response. The IBI variability was not able to distinguish between the levels in cases where the behavioral response could not distinguish between the levels. The EEG also indicated an increase in P300 latency which correlated with the increase in load and decrease in variability. Unfortunately no useful information was obtained from the MEG data due to head movement. These results indicate that there are significant changes in the physiological variables and the importance of providing a converging criterion.

Akselrod, et al 1981

S. Akselrod, D. Gordon, F. A. Ubel, D. L. Shannon, D. L. Berger, Power Spectrum Analysis of heart rate fluctuation: A quantitative probe of beat to beat cardiovascular control. Science 1981, v213, 220-221,

Aunon & McGillem 1975

Aunon, J. I. and McGillem, C.D. "Techniques for Processing Single Evoked Potentials," Proceedings of the San Diego Biomedical Symposium, San Diego, California, 1975.

Aunon 1978

Aunon, J. I., "Computer Techniques for the Processing of Evoked Potentials" Computer Programs in Biomedicine 8(1978) 243-255 Elsevier/North Holland Biomedical Press.

Aunon et al 1981a

Aunon, J. I., McGillem, C. D., and Childers, D. G., "Signal Processing Techniques in Evoked Potential Research: II-Averaging and Modeling Critical Reviews in Bioengineering, 5:323-367, 1981.

Aunon, et al 1982b

Aunon, J. I., McGillem, C. D., and O'Ponnel, R. D., "Comparison of Linear and Quadratic classification of event related potentials on the basis of their exogenous or endogenous components, Psychophysiology, 1982, 19, 531-538.

Aunon, et al 1984

J. I. Aunon, Chen, C. L., McGillem, C. D., Plonski, M. P., "Techniques for Analysis of Magnetic Evoked Potentials" AFAMRL-TR-84-008 Wright Patterson AFB, Dayton, Ohio 1984.

Begleiter, et al 1983

Begleiter, H., Porjesz, B., Chou, C., Aunon, J., "P3 and Stimulus Incentive Value," Psychophysiology, vol. 20, no. 1, pp. 95-101, Jan 83.

Bendat & Piersol 1971

Bendat, J. & Piersol, A., "Random Data: Analysis & Measurement Procedures," New York: Wiley-Interscience 1971,

Brazier 1984

Brazier, Mary, A. B. "Pioneers in the Discovery of Evoked Potentials," *Electroenceph. Clin. Neurophysical* Vol. 59, pp. 2-8, 1984.

Broadbent 1958

Broadbent, D. E., *Perception and Communication*, London: Pergamon Press 1958.

Broadbent 1971

Broadbent, D. E., *Decision & Stress*, New York: Academic Press, 1971.

Carlton&Kata 1980

Carlton, E., Kata, S., "Is Wrener Filtering an Effective Method of Improving Evoked Potential Estimation?" *IEEE Transactions on Biomedical Eng*, Vol BME-27 #4, April 1980.

Childers, et al 1981

Childers, D. G., Aunon, J. I., McGillem, C. D., "Signal Processing Techniques in Evoked Potential Research: III, Spectral Analysis Prediction and Extrapolation" *Critical Reviews in Bioengineering* 6: 133-175, 1981.

Childers, et al 1982

Childers, D., Bloom, P., Arroyo, A., Roucos, S., "Classification of Cortical Responses Using Features from Single EEG Records," *IEEE tran BME-29* #6, June 82.

Cohen et al. 1983

Cohen D. and Cuffin, B. Nerl "Demonstration of useful differences between magnetoencephologram and electroencephalogram." *Electroencephalography and Clinical Neurophysiology* 1983, 56:38-51.

Cohen 1985

Cohen, D., "Introduction: A 1984 Perspective on Biomagnetism in Biomagnetism: Applications & Theory edited by Weinberg, Stroind & Katila, Permagon Press: New York 1985.

Cuffin & Cohen 1977

Cuffin, B. N. and Cohen, D., Magnetic Fields of a Dipole in Special Volume Conductor Shapes, IEEE Trans. on BME-24, #4, July 1977.

Dawson 1947

Cerebral Responses to Electrical Stimulation at Peripheral Nerve in Man. J. Neurol. Neurosurg. Psychiat 10:134-140 1947.

DeBoer et al 1984

DeBoer, R., Karemaker, J., Strackee, J. Comparing Spectra of a Series of Point Event Particularly for Heart Rate Variability Data, IEEE Trans. on BME-31, #4, April 1984.

Donchin 1966

Donchin, E., "A Multivariate Approach to the Analysis of Average Evoked Potentials," IEEE, BME-13 #3, July 1966.

Donchin, et al 1973

Donchin, E., Kubory, M., Kutas, M., Johnson, R., Herning, R. I. Graded Changes in evoked response (P300) amplitude as a function of cognitive activity. Perception & Psychophysics, 1973, 319-324.

Donchin & Herning 1975

Donchin, E. R., Herning, R. I. "A simulation of the efficacy of stepwise discriminant analysis in the detection & comparison of event related potentials." Electroenceph. Clin., Neurophys. vol. 38, 1975 51-68.

Duncan-Johnson & Donchin 1981

Duncan-Johnson, C. C., Donchin, E. "The P300 Component of the Event-Related Brain Potential as an Index of Information Processing" *Biological Psychology* 14 (1982) 1-52 North-Holland Publishing Company

Firth 1973

Firth, R. A., Psychological factors influencing the relationship between cardiac arrhythmia and mental workload. *Ergonomics* 1973, vol. 16, 5-17.

Franks 1981

Franks, L. E., "Signal Theory," Stroudsburg PA.: Dowden & Culver, 1981.

Frasca 1983

Frasca, S., "Pulse Processes Analysis: Theory and Application", H.W. Schussler, Editor, Elsevier Science Publishers B. V. EURASIP 1983

Fukunaga 1972

Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press, N.Y.

Geddes & Baker 1975

Geddes, L. A. and Baker, L. E., Principles of Applied Biomedical Instrumentation 2nd Ed. Wiley: 1975.

Gevins, et al 1979a

Gevins, A., Zertlin, G. Yingling, C., Doyle, J. C., Dedon, M. F., Schaffer, R. E., Roumasset, J. T., Yeager, C. L., "EEG Patterns during cognitive tasks I: Methodology & analysis of complex behaviors," *Electroenceph & clinical Neurophysiol*, 1979 47: 693-703.

Gevins, et al 1979b

Gevins, A. S., Zeitlin, G-M., Doyle, J. C., Schatter, R. E., Calloway, E., "EEG Patterns during cognitive Task II Analysis of Controlled Tasks," *Electroencephlography & Clinical Neurophysiology* 1979, 704-710

Gevins, et al 1981

Gevins, A., Doyle, J., Cutillo, B., Schaffer, R., Tannehill, R., Ghannam, J., Gilcrease, V., and Yeager, C. New Method reveals dynamic patterns of correlation of human brain electrical potentials during cognition, *Science*, 1981, 213, 918-922.

Gevins 1980

Gevins, A. "Pattern Recongition of Human Brain Electrical Potentials," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. PAMI-2, #5 Sept. 1980.

Gevins 1984a

Gevins, Alan S., *Analysis of the Electromagnetic Signals of the Human Brain: Milestones, Obstacles and Goals.* *IEEE Trans, BME* 31(2) Dec. 1984.

Gopher & Braune 1984

Gopher, Daniel & Braune, R., "On the psychophysics of workload: why bother with subjective measures? Special Issue - Aviation Psychology - Human Factors, 1984, Vol. 26(5) 519-532.

Guyton 1981

Guyton, A. "The Normal Electrocardiogram." *Textbook of Medical Physiology.* Chapters 14, 15, 17. W.B. Saunders Company. 1981

Halliday, et al 1985

Halliday, D. L., McGillem, C. D., Westerkamp, J., Aunon, J. I., "Optimal & Suboptimal Feature Selection for Classification of Evoked Brain Potentials," IEEE Transactions on Systems, Man & Cybernatics, SMC-15, #3, May 1985.

Hart & Bortolussi 1984

Hart, S., Bortolussi, M., "Pilot Errors as a Source of Workload," Human Factors 1984, 26(5)-545-556

Herman & Kantowitz 1970

Herman, L. M. and Kantowitz, B. H., "The psychological refractory period: Only half the double stimulation story?" Psychological Bulletin, 1970, 73 74-88.

Heslegrave et al 1979

Heslegrave, R., Ogilvie, J., Furedy, J. Measuring Baseline-Treatment Differential in Heart Rate Variability: Variance versus Successive Difference Mean Square and Beats per Minute versus Interbeat Interval. Psychophysiology. Vol. 16 #2 1979

Israel, et al 1980

Israel, J. B., Wickens, C. P., Chesney, G. L., & Donchin, E. The event related brain potential as an index of display-monitoring workload. Human factors, 1980, 22, 211-224.

Jasper 1958

Jasper, Herbert H., "The Ten Twenty Electrode System of the International Federation," Electroen. Clin. Neuroph., vol. 10, pp. 371-375, 1958.

John, et al 1973

John, E. R., Walker, P., Camood, D., Rush, M., Gehrmaan, J. Factor Analysis of Evoked Potentials, Electroencephology & Clinical Neurophysiology 1973, 34: 33-43.

John 1977

John, E. "Functional Neuroscience", vol. II Neurometrics: Clinical Application of quantitative electrophysiology., Hillsdale, N.J.: Lawrence Erlbaum & Associates 1977

Kahneman 1973

Kahneman, D., Attention and Effort. Prentice Hall: Englewood Cliffs, N.J. 1973

Kalsbeck & Ettema 1963

Continuous Recording of Heart Rate & measurement of Perceptual load Ergonomics, 1963, vol. 6, 306-307.

Kalsbeek 1973

Kalsbeek, J. W. H., Do You Believe in Sinus Arrhythmia Ergonomics, 1973, vol. 16. 99-104.

Kantowitz & Knight 1974

Kantowitz, B. H. and Knight, J. L., "Testing Tapping Timesharing," Journal of Experimental Psychology 103, 331-336.

Kantowitz 1974

Kantowitz, B. H., Double Stimulation. In: B. H. Kantowitz (ed.), Human information processing: Tutorials in performance & cognition. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Kantowitz & Knight 1976

Kantowitz, B. H. and Knight, J. L., "Testing Tapping Timesharing: II Auditory Secondary Task," Acta Psychologica 1976, 40, 343-362.

Kantowitz 1985a

Kantowitz, B. A., "Stages & Channels in human information processing: A limited Analysis of Theory and Methodology. Journal of Mathematical Psychology #2, June 1985.

Kantowitz 1985b

Kantowitz, B. H., "Brocybernetic Analysis of a hybrid workload model" AFSOR T.R. BRA.85-10, May 1985, Bolling AFB.

Karis & Donchin, et al 1983

Karis, D., Chesney, G. L., Donchin, E. "'... 'twas ten to one; And yet we ventured...': P300 and Decision Making Psychophysiology Vol 20, No. 3, 1983.

Katila 1981

Katila, T., "Instrumentation for Biomedical Applications, in Biomagnetism, edited by Erne, Hahlbohn, Cubbig 1981 Walter de Gruyter & Co., Berlin, New York.

Krol & Opmeer 1971

Krol, J. P. & Opmeer C. H. J. M., Fysiologische variabelen sinus - aritmie, hart frequentie en ademfrequentie bij parachutisten, Report from Laboratory for Ergonomic Psychology TNO

Kutas & Donchin, et al 1977

Kutas, M., McCarthy, G., Donchin, E. "Augmenting Mental Chronometry: The P300 as a Measure of Stimulus Evaluation Time" 1977 Science vol 197 792-795

Kay & Marple 1981

Kay, S., Marple, S. "Spectrum Analysis - A Modern Perspective", Proc. of the IEEE, Vol. 69, No. 11, November 1981

Lay 1982

Lay, Steven, Convex sets and their applications, Wiley Interscience: New York 1982.

Lewis et al 1985

Lewis, G., Blackburn, M. Nartoh, P., Metcalfe, M., "Few-Trial Evoked Field Stability using the DC squid" in Weinberg Stroink, Katila (eds) *Bromagnetism: Applications & Theory*, Permagon Press: New York 1985.

Kwaadsteniet 1982

Kwaadsteniet, J. W., "Statistical Analysis and Stochastic Modeling of Neuronal Spike-Train Activity" *Mathematical Biosciences* 60:17-71 1982.

Magliero & Donchin 1981

Magliero, A., Bashore, T. R., Coles, M. G. H., Donchin, E. "On the dependence of P300 Latency on Stimulus Evaluation Processes" *Psychophysiology* Vol 21, No 2, 1984.

McCarthy & Donchin 1981

McCarthy G., and Donchin, E., "A metric for thought; A comparison of P300 latency & reaction time." *Science*, 1981, 211, 77-80.

McClellan, et al 1979

McClellan, J., Parks, T., Rabiner, L., "FIR Linear Phase Filter Program" in *Programs for Digital Signal Processing.*, ed: Digital Signal Processing Committee, New York: IEEE Press 1979

McGillem & Aunon 1977

McGillem, C. D. & Aunon, J. I., "Measurements of Signal Components in Single Visually Evoked Brain Responses." *IEEE transaction on BME*, Vol. 24, No. 3, 232-241, 1977.

McGillem et al 1981

McGillem, C. D., Aunon, J. I., & Childers, D. G., "Signal Processing in Evoked Potentials Research: I - Filtering Techniques and Pattern Recognition Methods," *Critical Reviews in Bioengineering*, vol. 6, 225-265, 1981.

McGillem & Aunon 1985a

McGillem, C. D., Aunon, J. I., "Detection, Estimation & Multidimensional Processing of Single Evoked Potentials" AFOSR-F49620-83-1C-0031, August 1985.

McGillem, et al 1985b

McGillem, C. D., Aunon, J. I., Pomalazu, C. A., "Improved Waveform Estimation Procedures for Event Related Potentials," IEEE Transactions on BME, BME-32, #6, June 1985.

Mohn 1976

Mohn, R. K., "Suggestions for the Harmonic Analysis of Point Process Data", Computers and Biomedical Research Vol 9, 521-530, 1976.

Moray, et al 1979

Moray, N., Johanssen, J., Pew, R. D., Ramusssm, J., Sanders, A. F. and Wickens, C. D. (1979). Report of the experimental psychology group. In N. Moray (Ed) Mental Workload, its theory and measurement. New York: Plenum, from #25 ref list.

Moser 1984

Moser, J., "Classification & Detection of Single Evoked Brain Potentials using time-frequency amplitude features" in Ph.D thesis Purdue University 1984.

Mulder et al 1973

Mulder, G., W. R. E. H. Mulder, Hajonides Van Der Meulen, "Mental Load and the Measurement of Heart Rate Variability" Ergonomics 1973, vol. 16 #1,69-83.

Myers et al 1986

Myers, G. A., Martin, G. J., Magid, N. M., Barnett, P. S., Schaad, J. W., Weiss, J. S., Lesch, M., Singer, D. H., "Power Spectral Analysis of Heart Rate Variability in Sudden Cardiac Death: Comparison to Other Methods" IEEE Transactions on Biomedical Engineering, Vol. BME-33, NO. 12, December 1986. sp

Naatanen 1975

Naatanen, R., "Selective Attention & Evoked Potentials -- A critical Review" *Biological Psychology*, vol. 2, pp. 237-307, 1975.

Nagy 1968

Nagy, G., "State of the art in pattern recognition," *Proc. IEEE*, 56-5, 836, 1968.

Okada et al 1981

Okada, Y. C., Kaufman, L., Brenner, D., Williamson, S. J. "Application of a SQUID to measurement of somatically evoked fields: transient responses to electrical stimulation of the medium nerve. In Erne, Hahlbohm, Lubbig (eds) *Biomagnetism*, Walter de Gruyter & Co: Berlin, New York 1981.

Okada 1983

Okada, Y. C., *Evoked Magnetic Fields*, pp. 399-468. In S. S. Williamson, G. L. Romani, L. Kaufman, and I. Modena (Eds), *Biomagnetism*, Plenum Press: N.Y. 1983.

Pachella 1974

Pachella, Robert G., "The interpretation of reaction time in information-processing research" in B. H. Kantowitz (ed), *Human Information Processing: Tutorials in Performance and Cognition*. Hillsdale, N.J. Lawrence Erlbaum Associates.

Peterka 1978

Peterka, R. J., Sanderson, A. C., O'Leary, D. P, "Practical Considerations in the Implementation of the French-Holden Algorithm for Sampling Neuronal Spike Trains", *IEEE Transactions on Biomedical Engineering*, Vol. BME-25, NO. 2, March 1978.

Porges & Bohrer 1980

New Time-Series Statistic for Detecting Rhythmic Co-Occurrence *Psychological Bulletin*, 1980, vol. 88, 580-587 in freq domain.

Priestley 1974

Priestley, M. B., Rao, Subbo, T., Tong, Howell, "Applications of Principal Component Analysis and Factor Analysis in the Identification of Multivariable Systems," IEEE trans on Automatic Control, Vol. AC-19, #6, Dec. 1974.

Pritchard 1981

Pritchard, W. S., "Psychophysiology of P300" "Psychological Bulletin" 1981, Vol. 89, 506-540.

Rolfe 1973

Symposium on Heart Rate Variability," Ergonomics, Vol. 16, Jan 1973.

Rompelman et al 1982

Rompelman, Otto, Snijders, Jos B.I.M, Van Spronsen, Cees J., "The Measurement of Heart Rate Variability Spectra with the Help of a Personal Computer", IEEE Transactions on Biomedical Engineering, Vol BME-29, NO. 7, July 1982.

Rosler & Monzey 1981

Rosler, F., Monzey, D., "Principal Components and Varimax-Rotated Components in Event Related Potential Research: Some remarks on their interpretation," Biological Psychology 13 (1981) 3-26.

Ruchkin et al 1975

Ruchkin, D., Sutton, S. & Tueting, P., "Emitted & Evoked P300 Potentials & Variations in Stimulus Probability," Psychophysiology, vol. 12, pp. 591-595, 1975.

Sachs 1984

Sachs, L. "Applied Statistics - A Handbook of Techniques," 2nd edition. New York: Springer - Verlag, 1984

Sarwinski 1977

Sarwinski, R. E., Superconducting Instruments, Cryogenics, Dec 1977, pp. 671-679.

Sayers 1973

Sayers, B. MCA., "Analysis of Heart Rate Variability," Ergonomics, 1973, Vol. 16, #1, 17-32.

Sayers 1975

Sayers, B. MCA., Physiological Consequences of Informational Load and Overload. Research in Psychophysiology. Edited by Venables & Christie, John Wiley & Sons, 1975

Sencaj et al 1979

Sencaj, R. W., Aunon, J. I., McGillem, C. D., "Discrimination among visual stimuli by classification of their single evoked potentials" Med & Biol Eng & Comput., Vol. 17, 1979.

Sharrt & Salvendy 1982

Sharrt, J. and Salvendy, G., "External and Internal attentional environments: II. Reconsideration of the relationship between sinus arrhythmia and information load," Ergonomics, 1982 (Feb) Vol. 25(2), 121-132.

Squires & Donchin 1976

Squires, K. C., Donchin, E. "Beyond Averaging: The use of Discriminant Functions to Recognize Event Related Potentials Elicited by Single Auditory Stimuli" Electroencephalography and Clinical Neurophysiology, 1976, 41: 449-459, Elsevier Scientific Publishing Company, Amsterdam.

Stenberg 1969

Stenberg, S., "The discovery of processing stages. Extensions of Donders' method." Acta Psychologica, 30, 276-315.

Taylor 1976

Taylor, David, Stage Analysis of Reaction Time, Psychological Bulletin, Vol. 83, March 1976, 161-191.

Telford 1931

Telford, C. W. "The refractory phase of voluntary and associative responses." Journal of Experimental Psychology, 1931, 14, 1-36.

Townsend 1974

Townsend, J. T. "Issues and Models Concerning the Processing of a finite number of inputs," in Human Information Processing. Tutorials in Performance and Cognition. Ed. by Kantowitz, Lawrence Erlbaum Associates: Hillsdale, N. J., 1974.

Valentinuzzi & Geddes 1974

"The Central Component of the Respiratory Heart-Rate Response" Cardiovascular Research Center Bulletin, Vol. XII, No. 4, April - June, 1974

VanRotterdam 1970

"Limitation and Difficulties in Signal Processing by Means of the Principal-Components Analysis" IEEE Transactions BME, July 1970.

Wastell 1981a

Wastell, D. G., "PCA and Varimax Rotation: Some Comments on Rosler and Monzey," Biological Psychology 13 (1981) 27-29.

Weiner 1982

Weiner, J. S. "The Ergonomics Society -- the society's lecture 1982: The measurement of Human Workload." Ergonomics, 1982 V 25 (11) 953-965.

Westerkamp 1985

Westerkamp, J., "A Multichannel Time Varying Filter for Estimating Nonstationary Signals in Noise" in Phd thesis, Purdue University, 1985.

Williamson & Kaufman 1981

Williamson, J. J. and Kaufman, L., Biomagnetism, Journal of Magnetism and Magnetic Materials, 22 (1981) 129-201.

Wood & McCarthy 1984

Wood, C., McCarthy, G., "Principal Component Analysis of Event-Related Potentials: simulation studies demonstrate misallocation of variance across components" Electronic & Clinical Neuroph. 1984, 59:249-260.

Woody 1967

Woody, C. D., "Characterization of an adaptive filter for the analyses of variable latency neuroelectric signals," Med. Biol. Eng 5, 539, 1967.

Womack 1971

Womack, Baxter F., "The Analysis of Respiratory Sinus Arrhythmia Using Spectral Analysis and Digital Filtering" IEEE Transactions on Bio-Medical Engineering, Vol. BME-18, NO. 6, November 1971.

Yu & McGillem 1982

Yu, K., McGillem, C. D., "Optimum Filters for Estimating Evoked Potentials Waveforms," IEEE-BME, vol.30, 1983.

APPENDICES

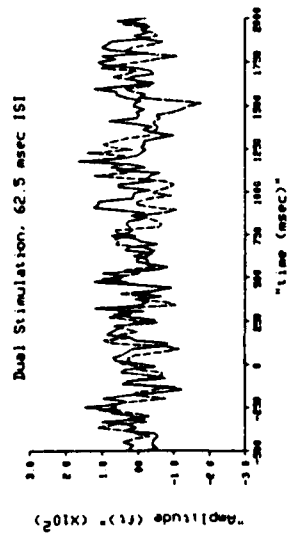
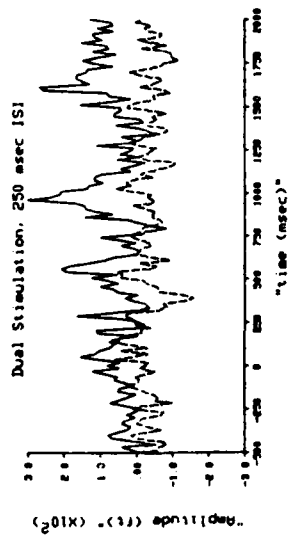
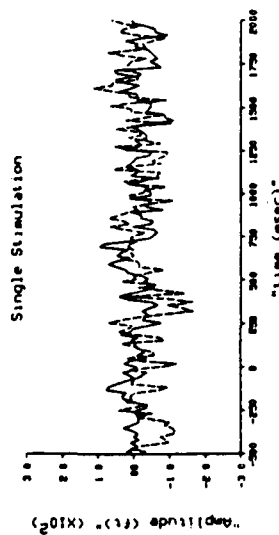
APPENDIX A

Averages and Power Spectral Estimates

MEG, 7cm above right ear
SUBJECT 9

AVERAGES

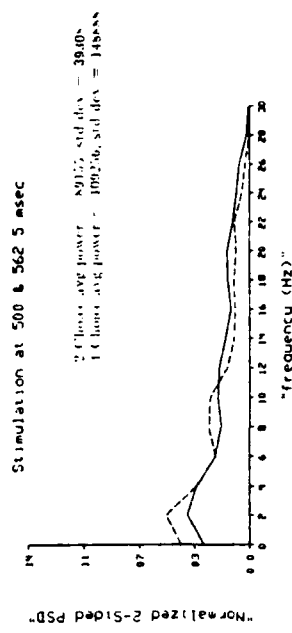
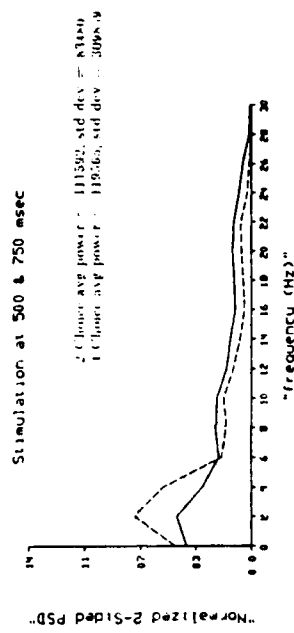
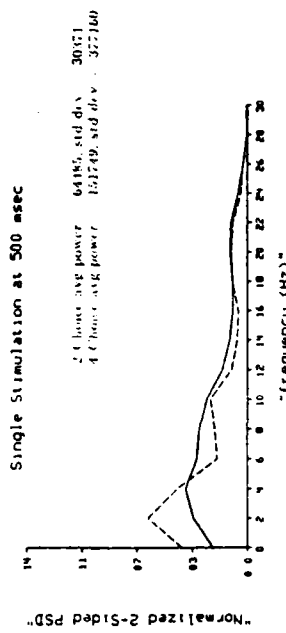
2 CHOICE CASE = solid
1 CHOICE CASE = dash



MEG
SUBJECT 9

Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

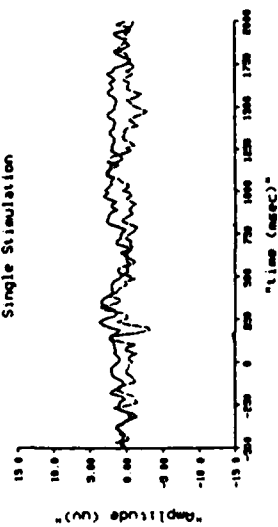


ELECTRODE P4
SUBJECT 9

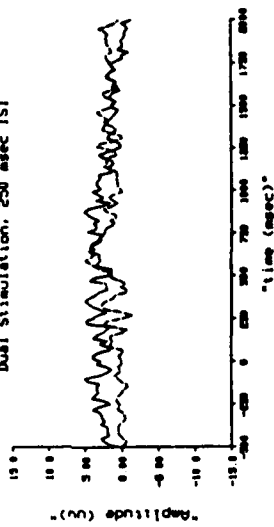
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

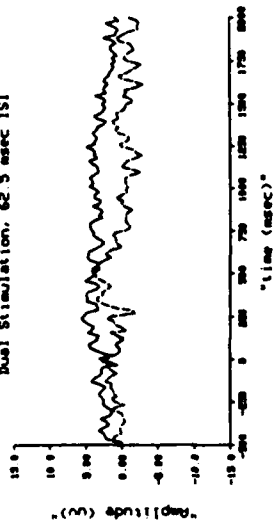
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

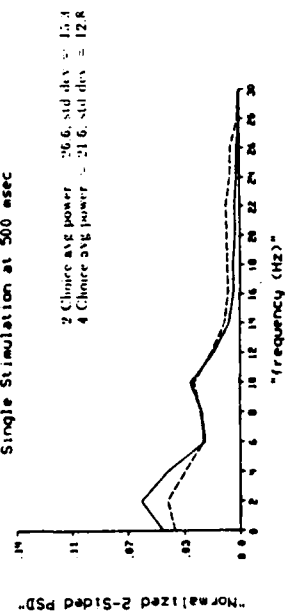


ELECTRODE P4
SUBJECT 9

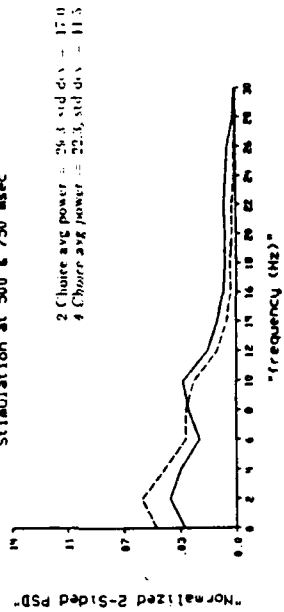
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dash

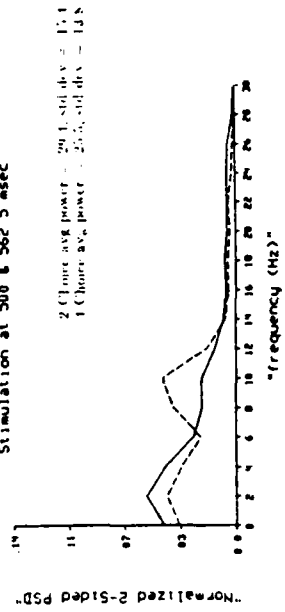
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec



ELECTRODE P3
SUBJECT 9

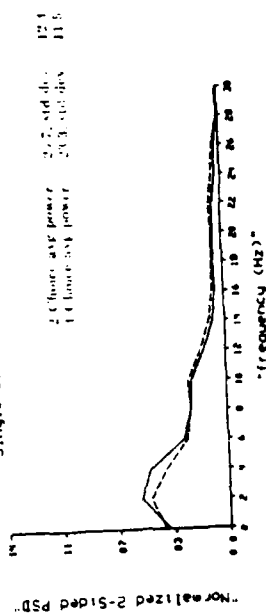
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

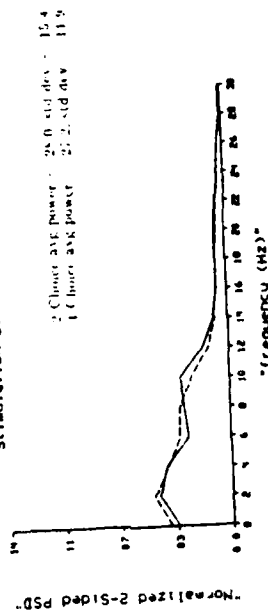
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

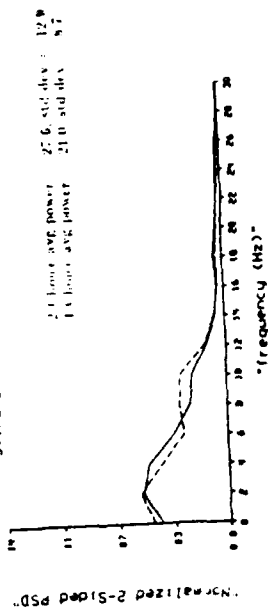
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec

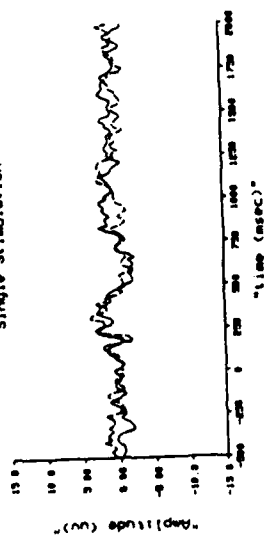


ELECTRODE P3
SUBJECT 9

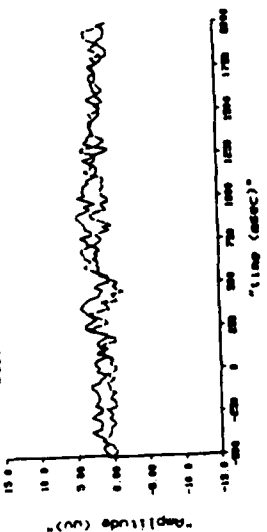
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

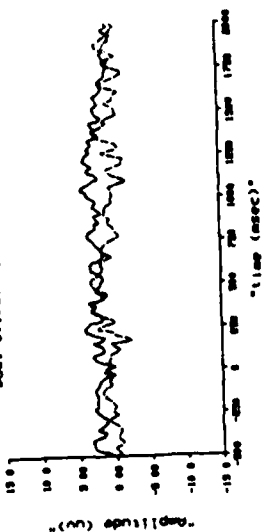
Single Stimulation



Dual Stimulation, 250 msec ISI



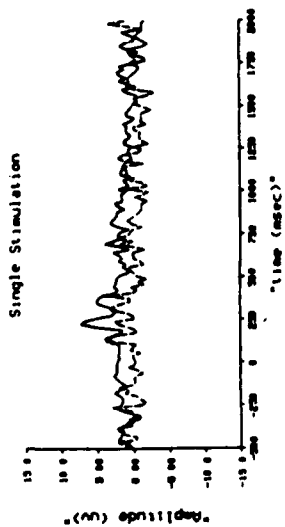
Dual Stimulation, 62.5 msec ISI



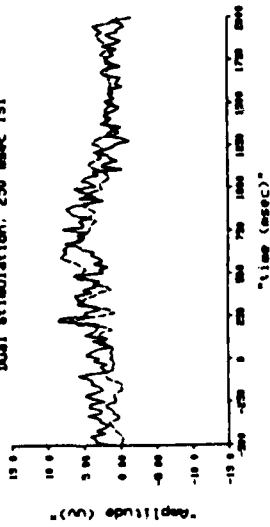
ELECTRODE C4
SUBJECT 9

AVERAGES

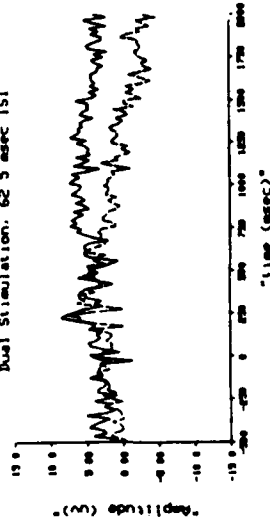
2 CHOICE CASE = solid
4 CHOICE CASE = dash



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

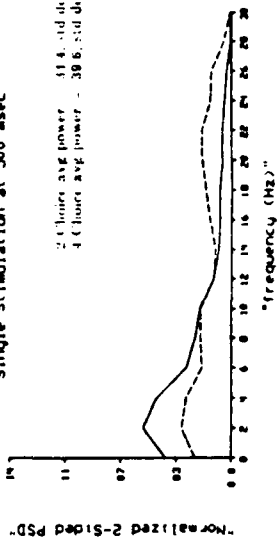


ELECTRODE C4
SUBJECT 9

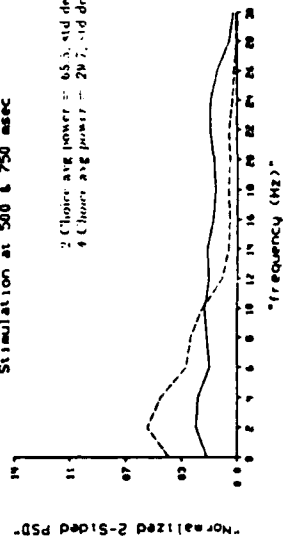
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

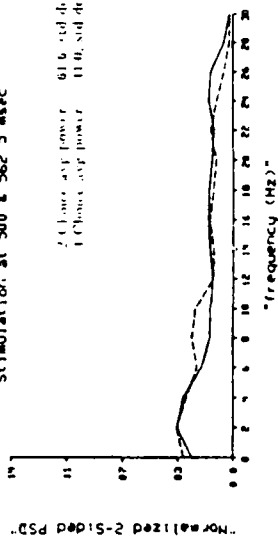
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec

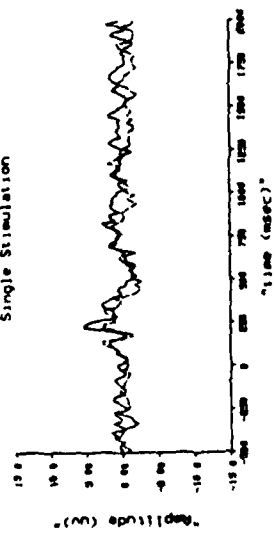


ELECTRODE C3
SUBJECT 9

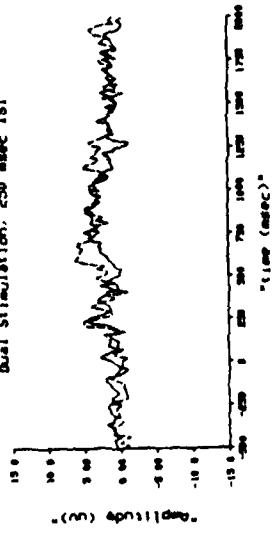
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

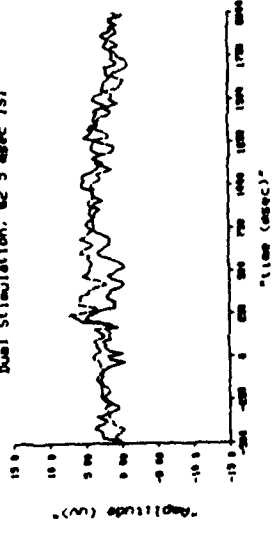
Single Stimulation



Dual Stimulation, 250 msec IS



Dual Stimulation, 62.5 msec IS

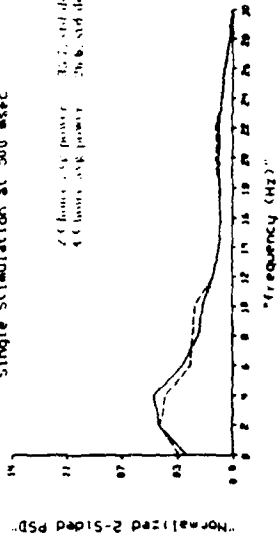


ELECTRODE C3
SUBJECT 9

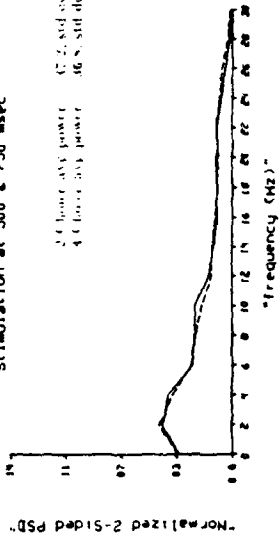
Normalized PSD
1/2 sec prestim

2 CHOICE CASE = solid
4 CHOICE CASE = dashed

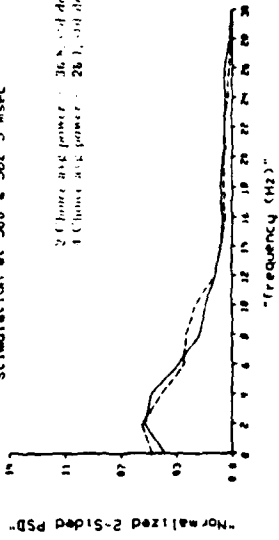
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec



2 Choice case power = 17.2
4 Choice case power = 12.5

2 Choice case power = 19.1
4 Choice case power = 18.6

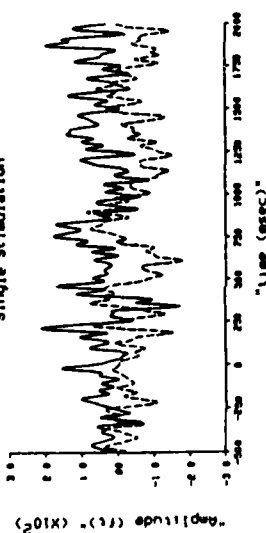
2 Choice case power = 25.7
4 Choice case power = 10.8

MEG: 7mm above right ear
SUBJECT: 8

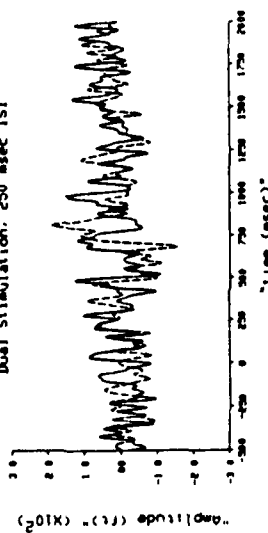
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

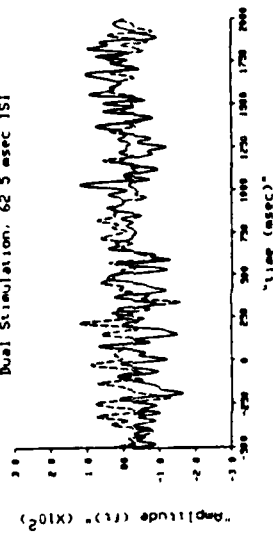
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

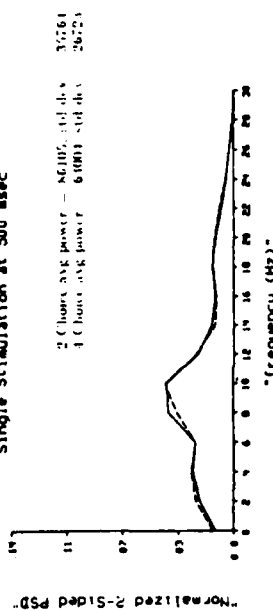


MEG
SUBJECT: 8

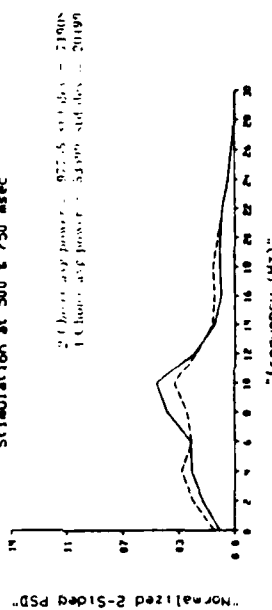
Normalized PSD
1/2 sec pre-stim

2 CHOICE CASE = solid
4 CHOICE CASE = dashed

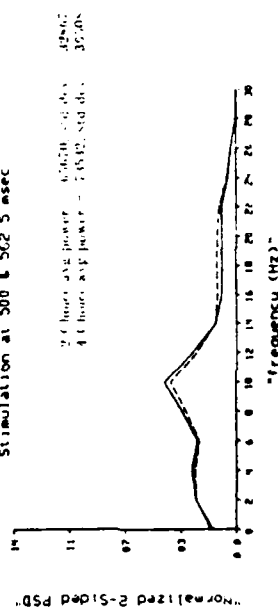
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec



ELECTRODE P4
SUBJECT 8

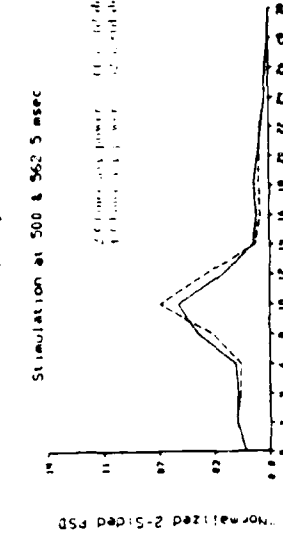
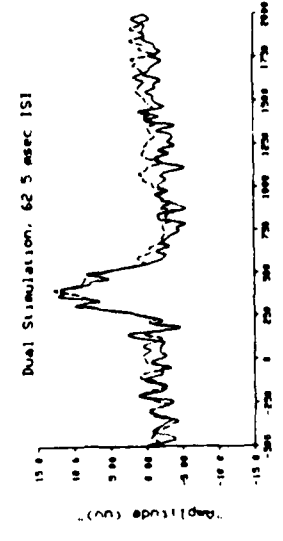
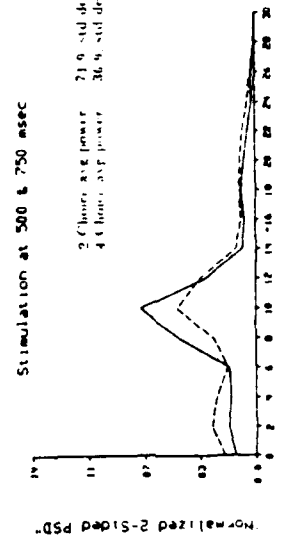
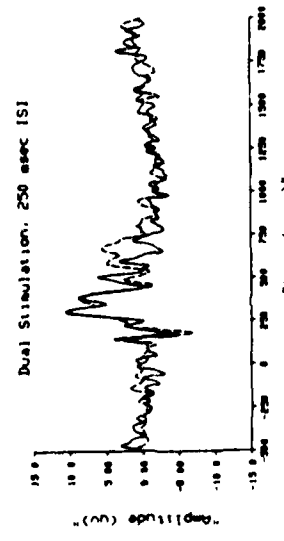
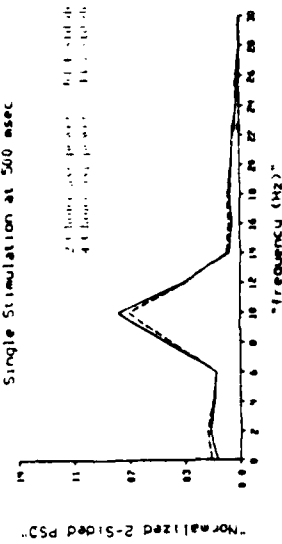
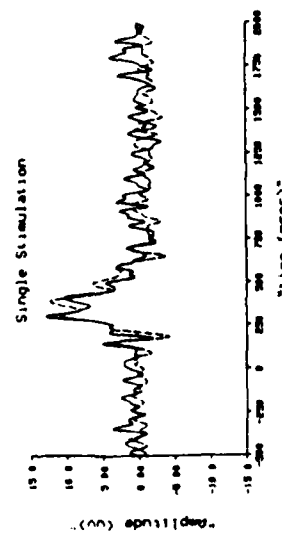
AVERAGES

2 CHOPPER CASE
4 CHOPPER CASE

Normalised PSD
1/2 sec pre-stim

2 CHOPPER
4 CHOPPER

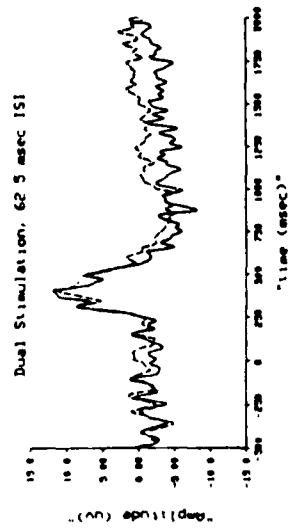
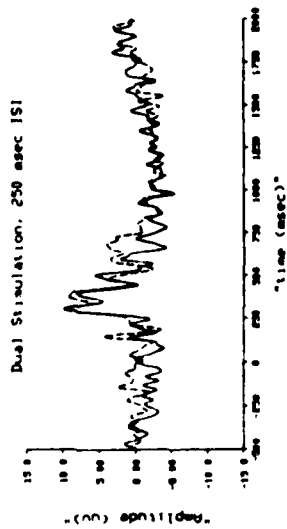
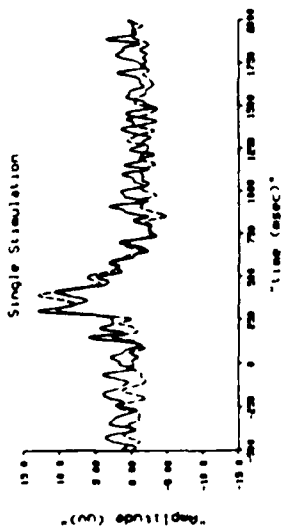
solid
dashed



EEG CHORD 13
SUBJECT 8

AVERAGES

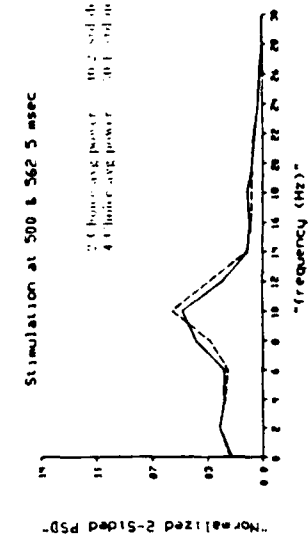
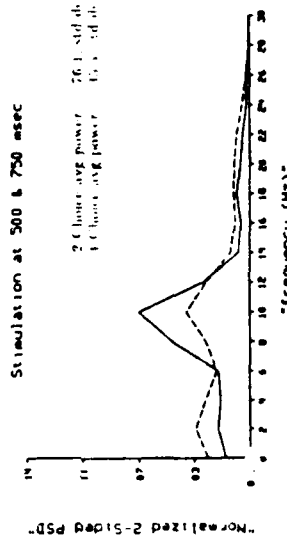
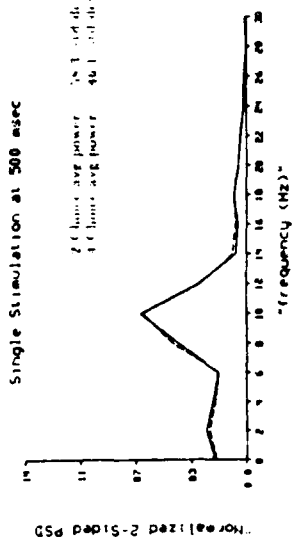
2 CHOICE CASE - solid
4 CHOICE CASE - dashed



EEG CHORD 13
SUBJECT 8

Normalized PSD
1/2 sec prestim

2 CHOICE - solid
4 CHOICE - dashed



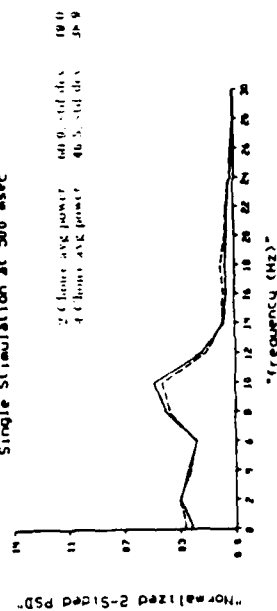
ELECTRODE C4
SUBJECT 8

AVERAGES
2 CHOICE CASE = solid
4 CHOICE CASE = dash

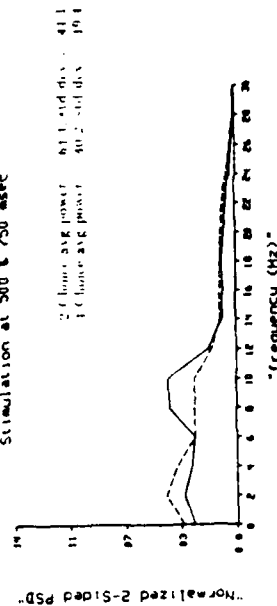
Normalized PSD
1/2 sec pre-stim

2 CHOICE CASE = solid
4 CHOICE CASE = dashed

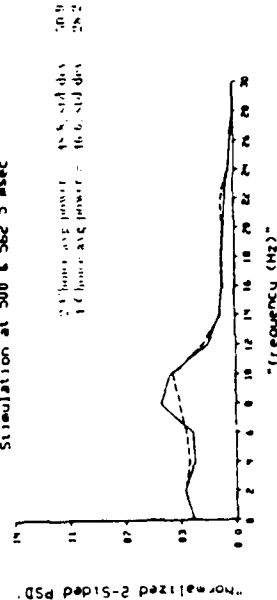
Single Stimulation at 500 msec



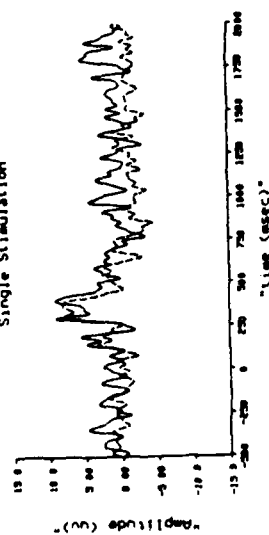
Stimulation at 500 & 750 msec



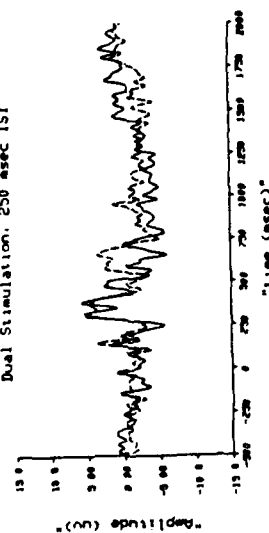
Stimulation at 500 & 562.5 msec



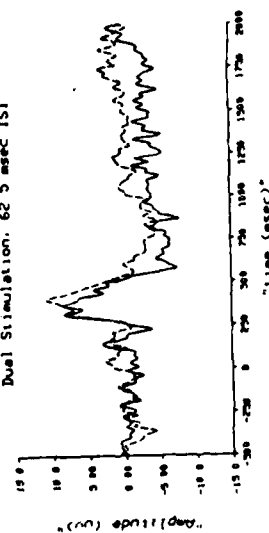
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI



ELECTRODE C3
SUBJECT 8

AVERAGES

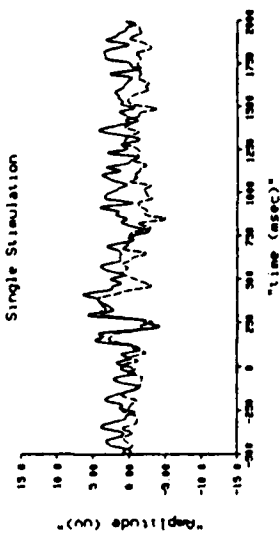
2 CHOICE CASE = solid
4 CHOICE CASE = dash

ELECTRODE C3
SUBJECT 8

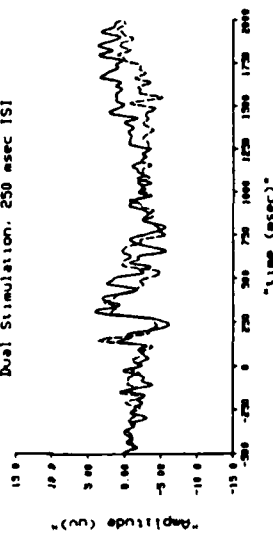
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

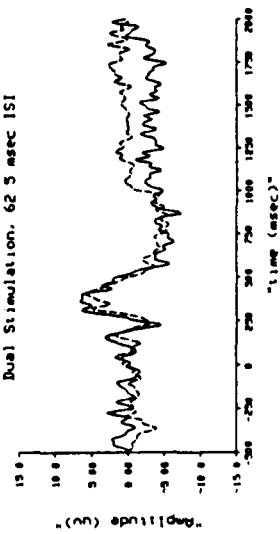
Single Stimulation



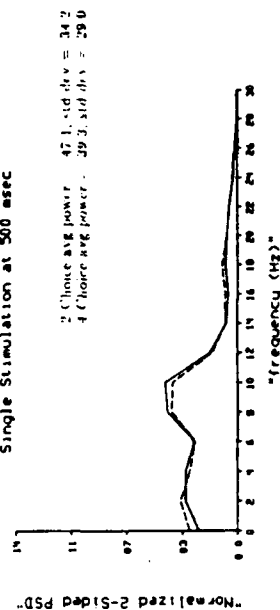
Dual Stimulation, 250 msec ISI



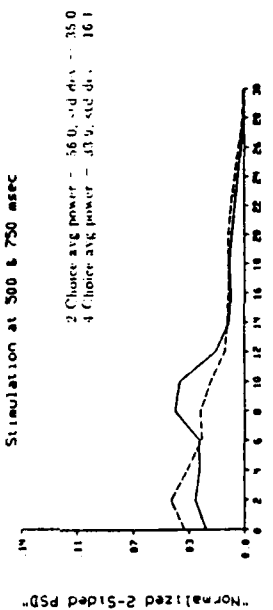
Dual Stimulation, 62.5 msec ISI



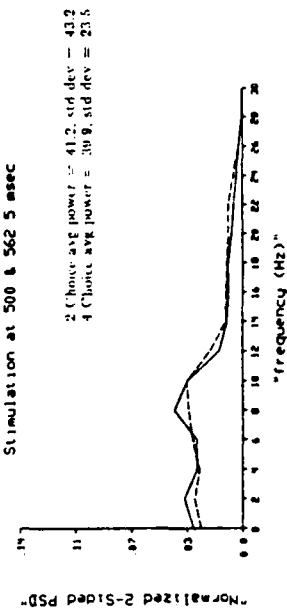
Single Stimulation at 500 msec



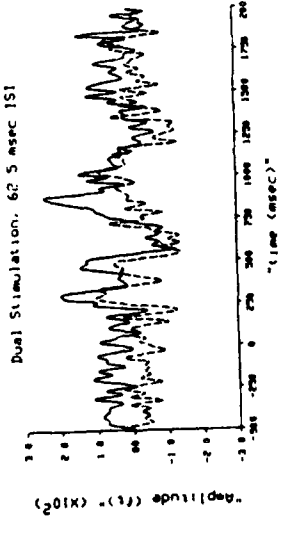
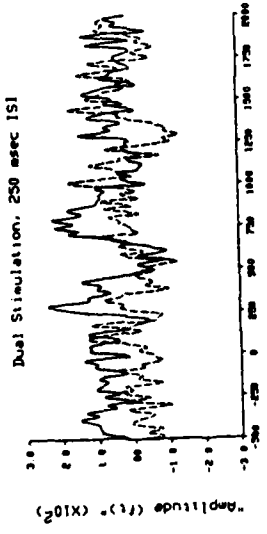
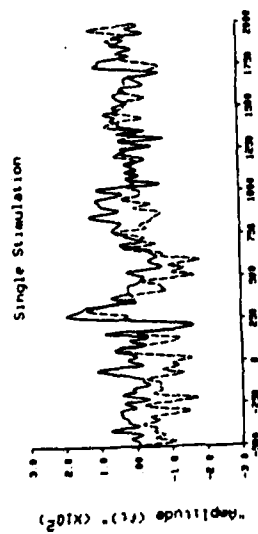
Stimulation at 500 & 750 msec



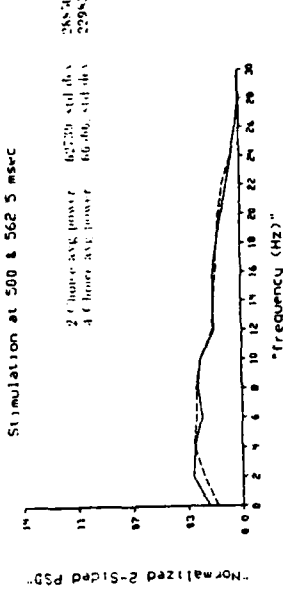
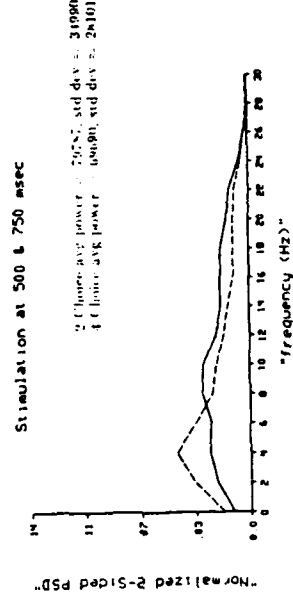
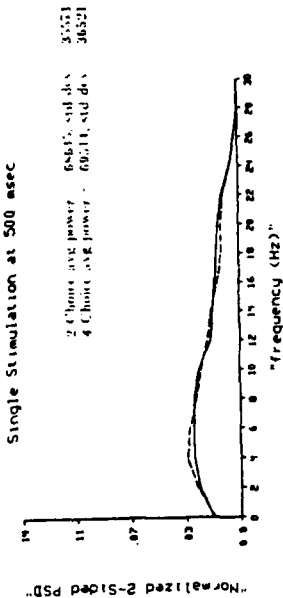
Stimulation at 500 & 562.5 msec



MEG
SUBJECT 7
AVERAGES
2 CHOICE CASE = solid
4 CHOICE CASE = dash



MEG
SUBJECT 7
Normalized PSD
1/2 sec prestim
2 CHOICE = solid
4 CHOICE = dashed

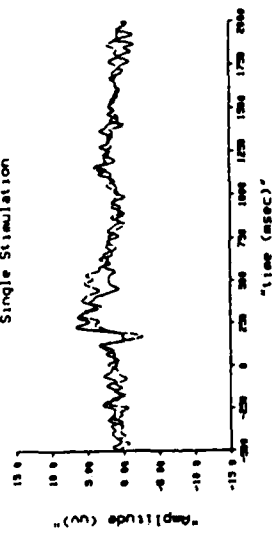


ELECTRODE P4
SUBJECT 7

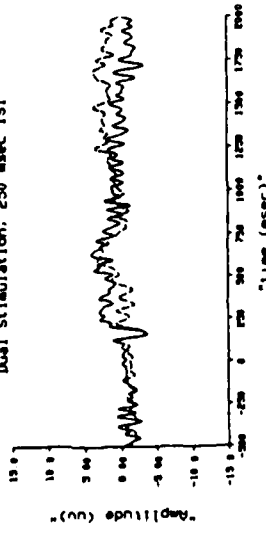
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

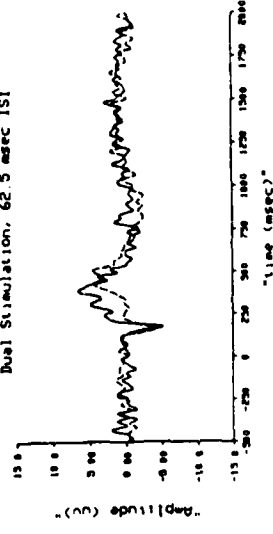
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

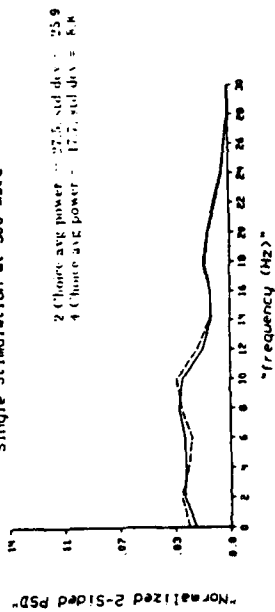


ELECTRODE P4
SUBJECT 7

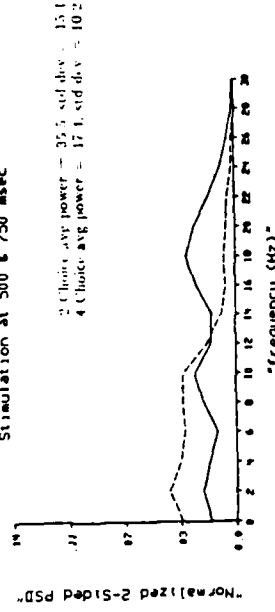
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

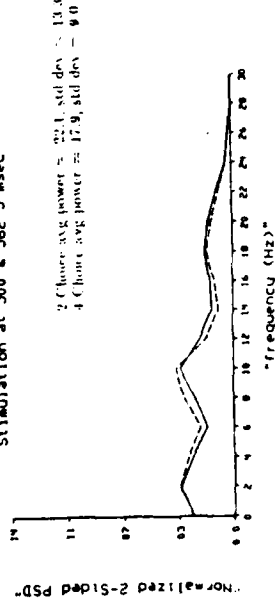
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec



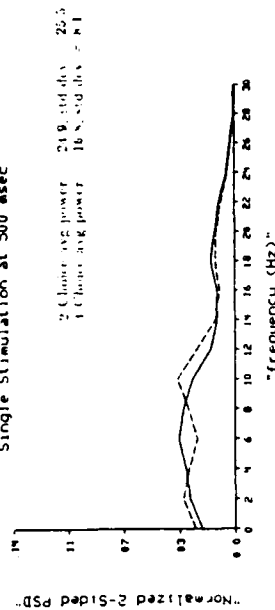
ELECTRODE: P3
SUBJECT: 7

AVERAGES: 2 CHOICE CASE - solid
4 CHOICE CASE - dash

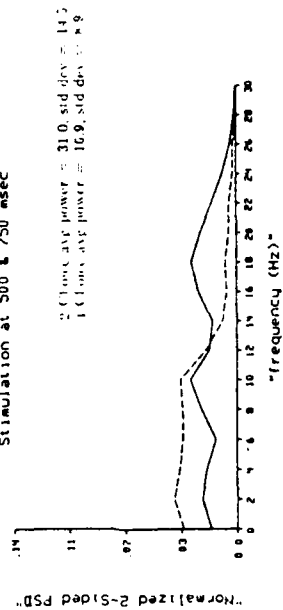
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dash

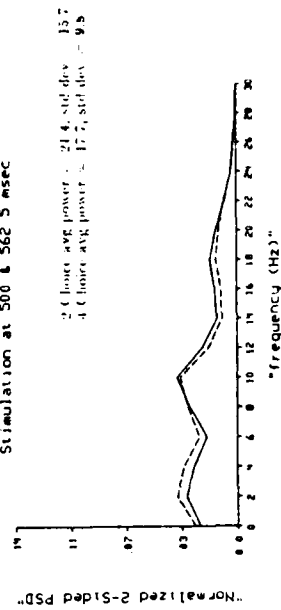
Single Stimulation at 500 msec



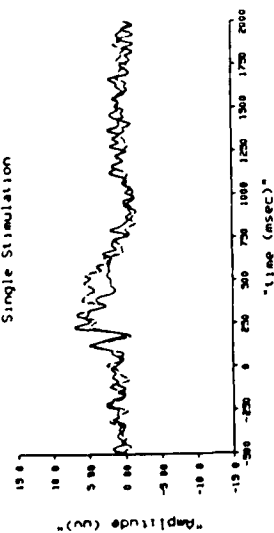
Stimulation at 500 & 750 msec



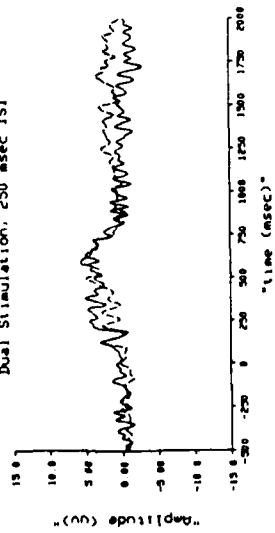
Stimulation at 500 & 562.5 msec



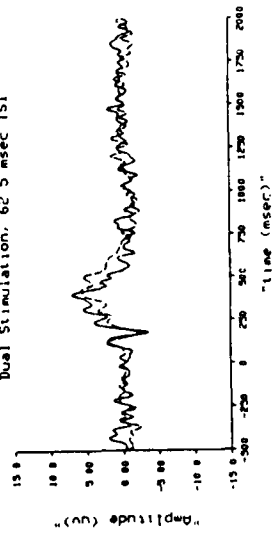
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI



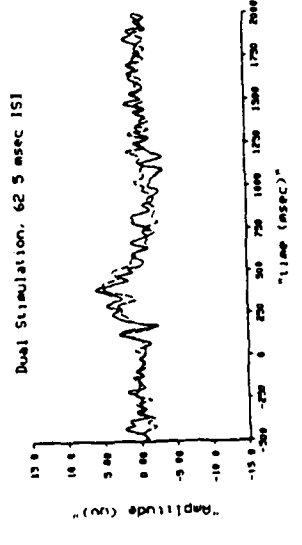
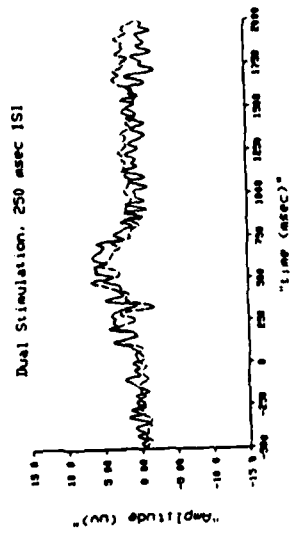
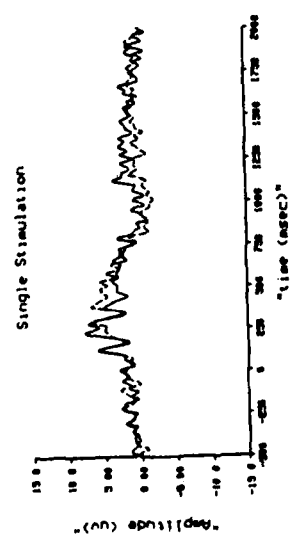
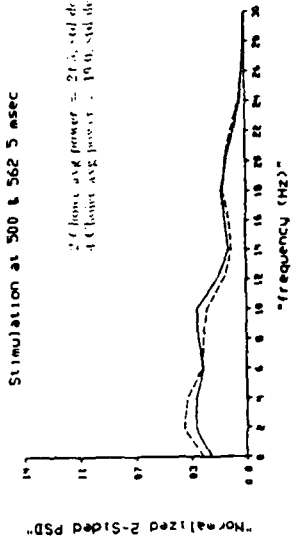
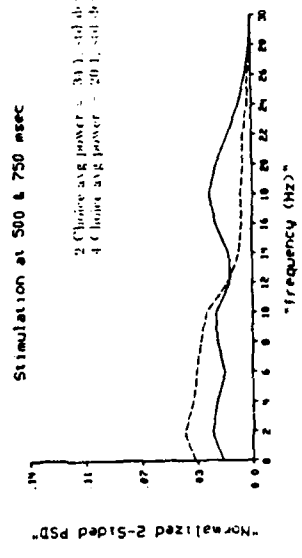
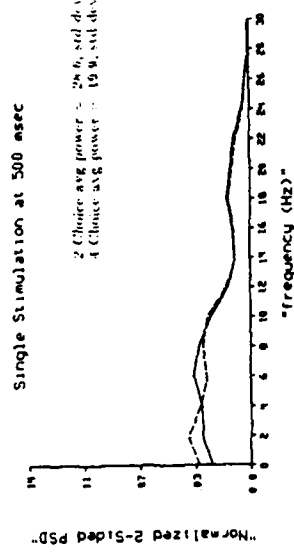
ELECTRODE C4
SUBJECT 7

AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

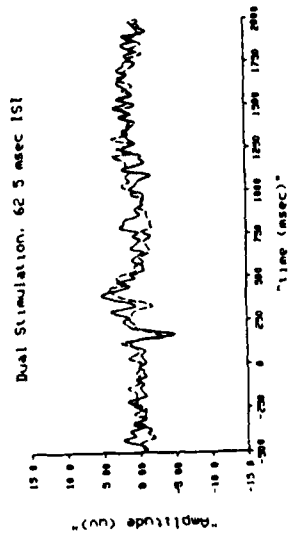
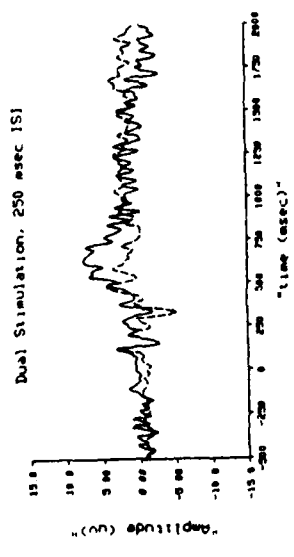
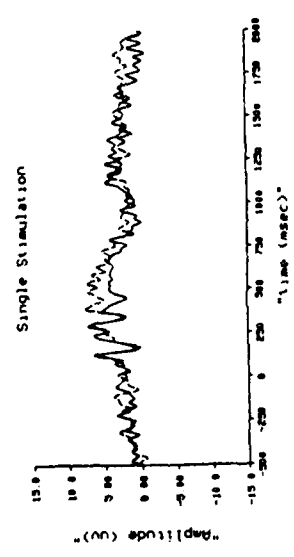
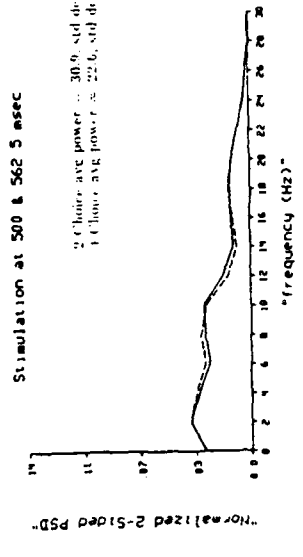
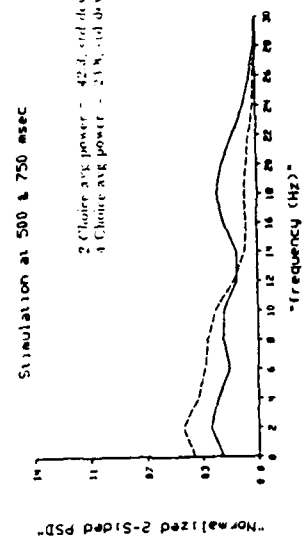
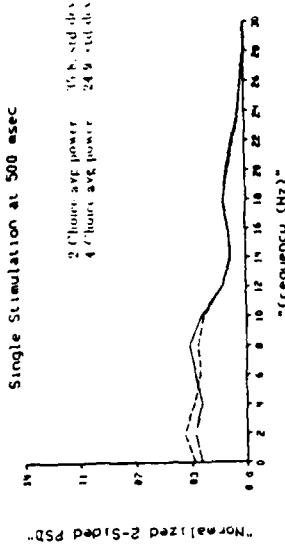


ELECTRODE C3
SUBJECT 7

AVERAGES
2 CHOICE CASE - solid
4 CHOICE CASE - dash

2 CHOICE - solid
4 CHOICE - dashed

Normalized PSD
1/2 sec prestim



MEG: 7cm above right ear
SUBJECT 6

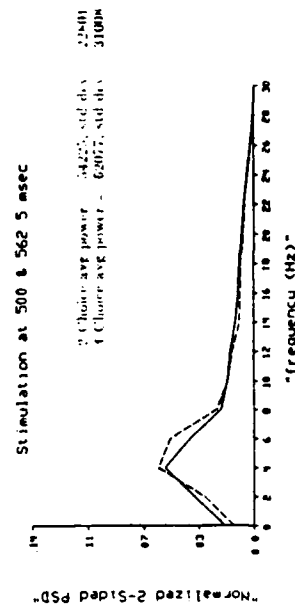
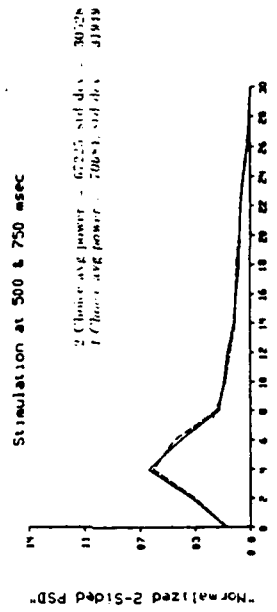
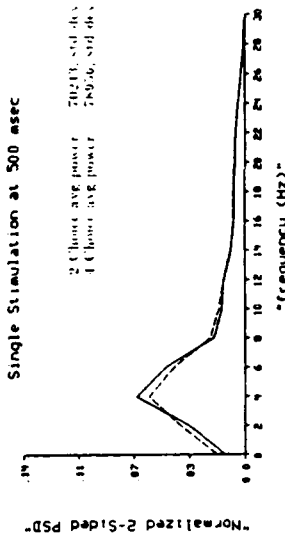
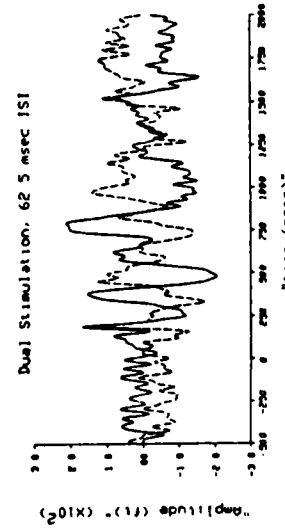
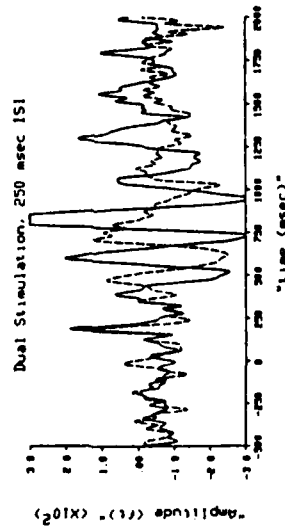
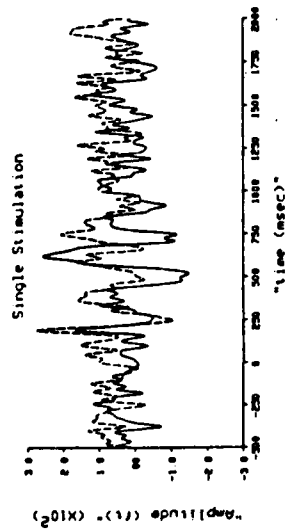
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

MEG
SUBJECT 6

Normalized PSD
1/2 sec prestim

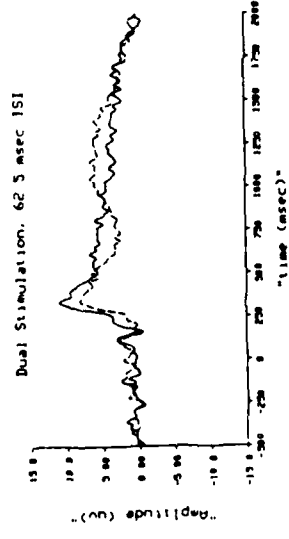
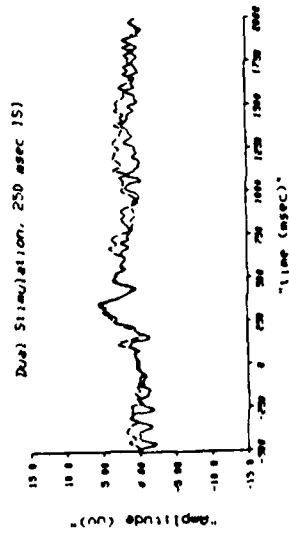
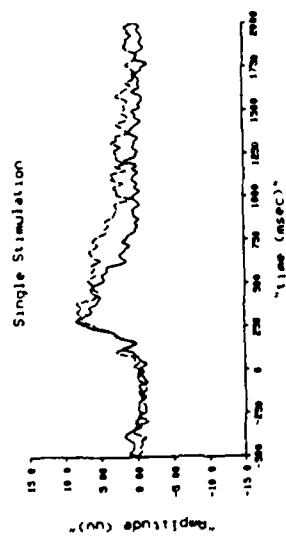
2 CHOICE = solid
4 CHOICE = dashed



ELECTRODE P4
SUBJECT 6

AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

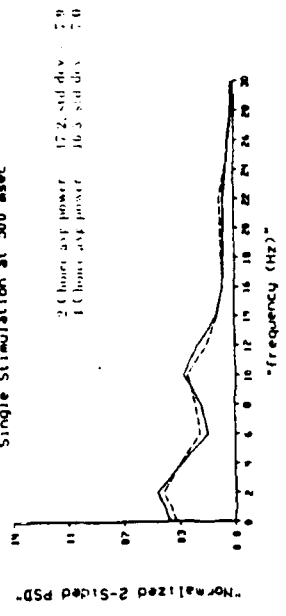


ELECTRODE P4
SUBJECT 6

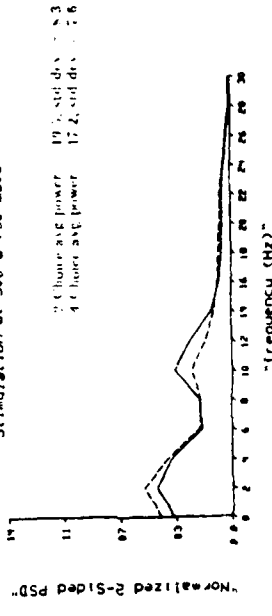
Normalized PSD
1/2 sec pre-tim

2 CHOICE = solid
4 CHOICE = dashed

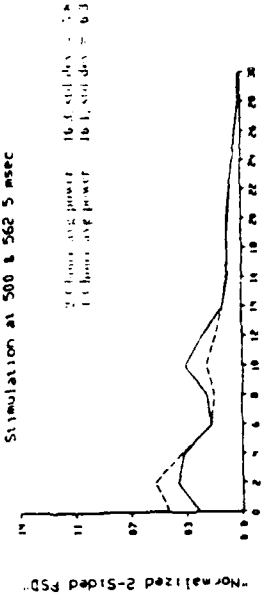
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



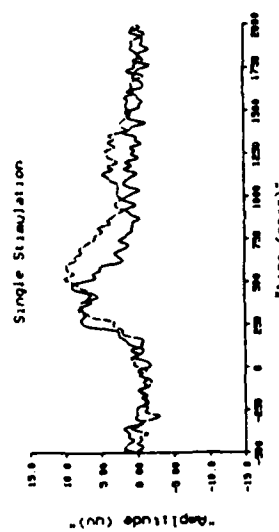
Stimulation at 500 & 562.5 msec



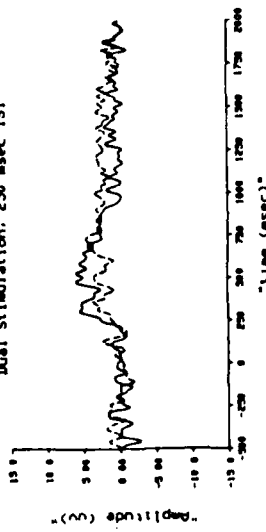
EELECTRODE P3
SUBJECT 6

AVERAGES

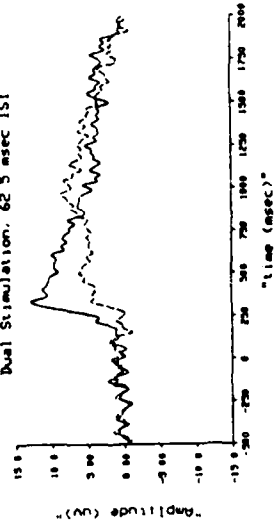
2 CHOICE CASE = solid
1 CHOICE CASE = dash



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

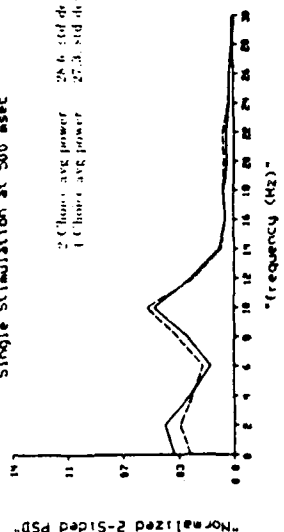


EELECTRODE P3
SUBJECT 6

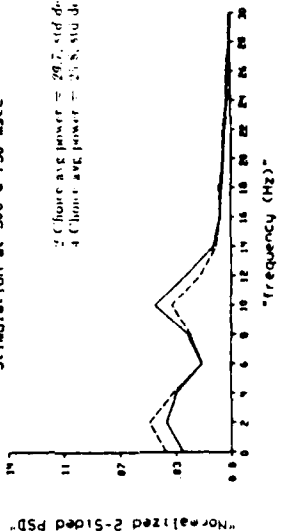
Normalized PSD
1/2 sec pre-stim

2 CHOICE = solid
4 CHOICE = dash

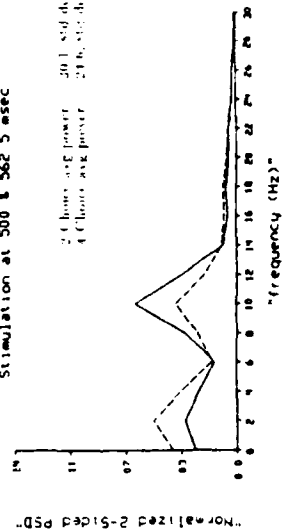
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec

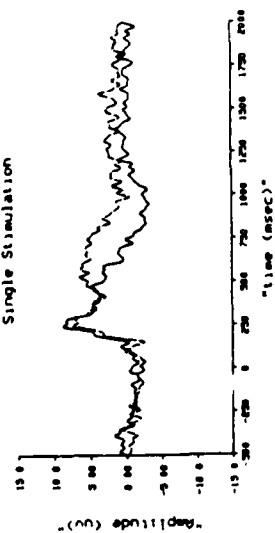


ELECTRODE C1
SUBJECT 6

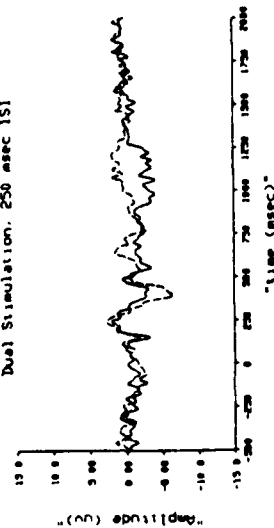
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

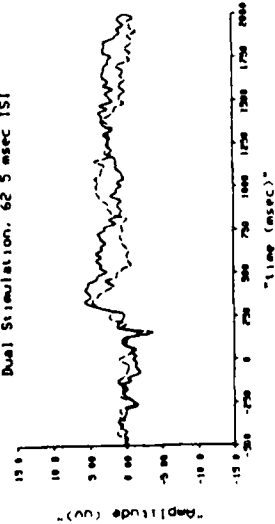
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

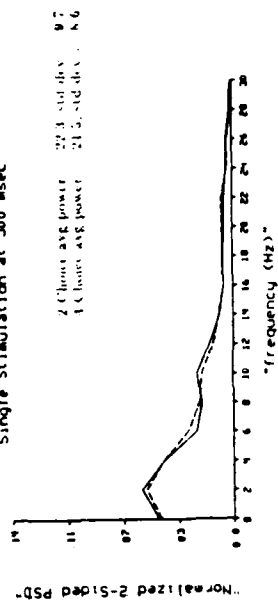


ELECTRODE C1
SUBJECT 6

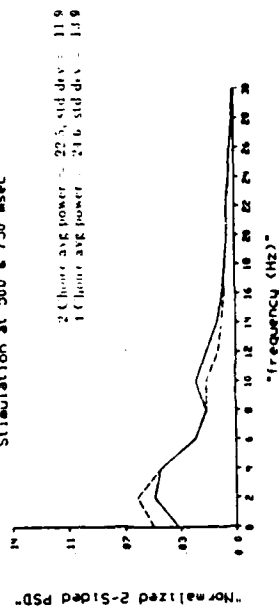
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dash

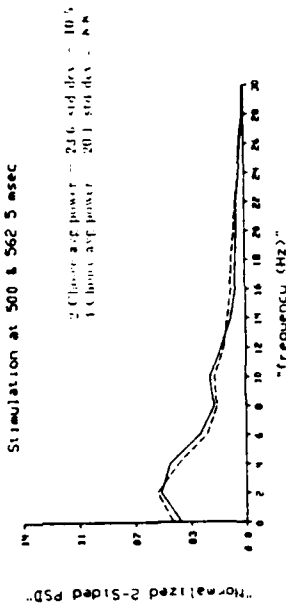
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



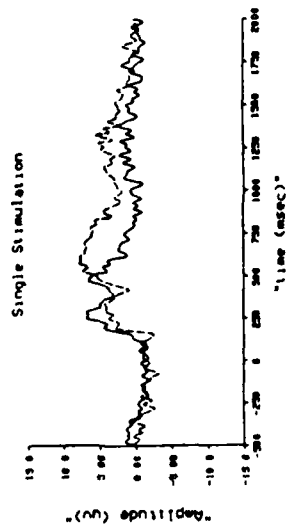
Stimulation at 500 & 562.5 msec



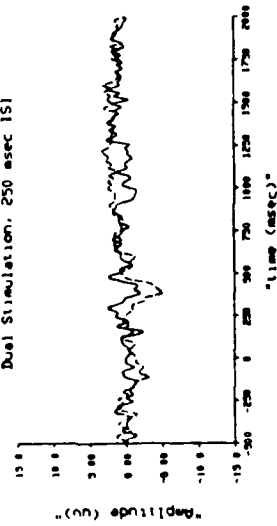
ELECTRODE: C3
SUBJECT: 6

AVERAGES

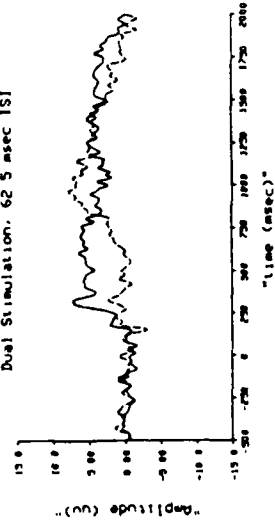
2 CHOICE CASE = solid
4 CHOICE CASE = dash



Dual Stimulation, 250 msec ISI



Dual Stimulation, 625 msec ISI

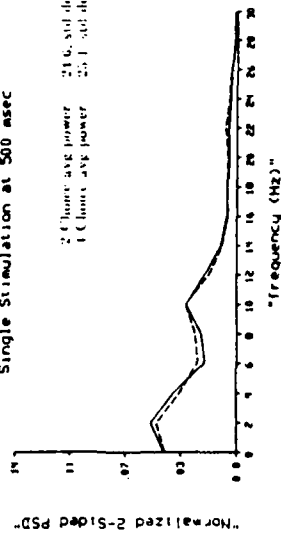


ELECTRODE: C3
SUBJECT: 6

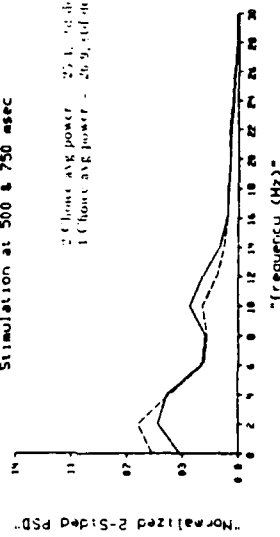
Normalized PSD
1/2 sec pre-tim

2 CHOICE = solid
4 CHOICE = dashed

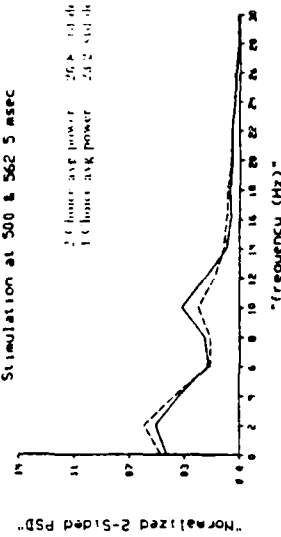
Single Stimulation at 500 msec



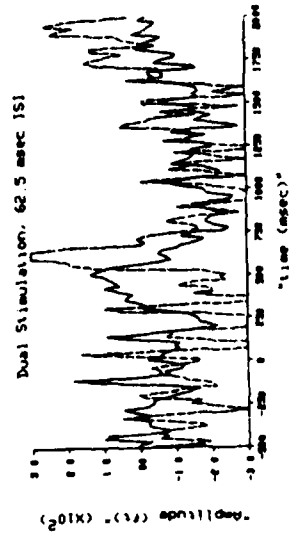
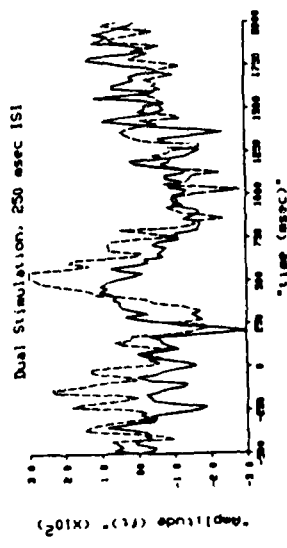
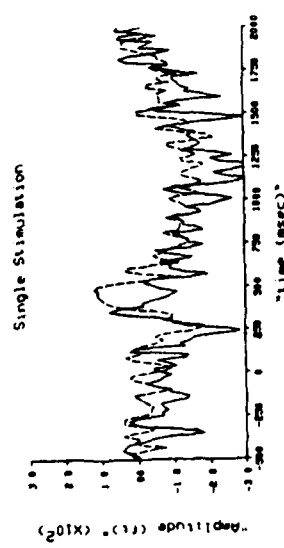
Stimulation at 500 & 750 msec



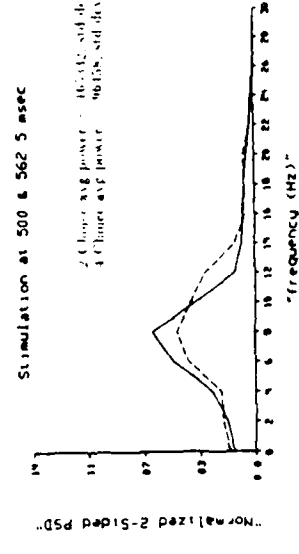
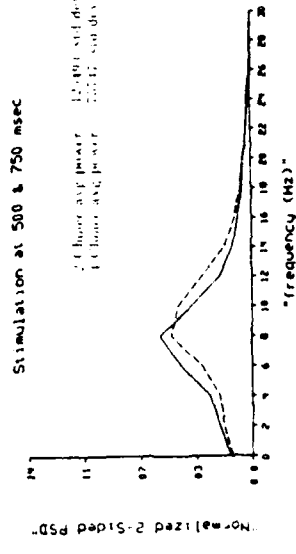
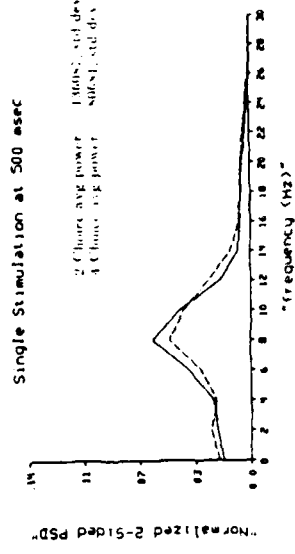
Stimulation at 500 & 562.5 msec



MUG, 7cm above right ear
 SUBJECT 5
 AVERAGES
 2 CHOICE CASE = solid
 4 CHOICE CASE = dash



MUG, 7cm above right ear
 SUBJECT 5
 Normalized PSD
 1/2 sec prism
 2 CHOICE = solid
 4 CHOICE = dashed

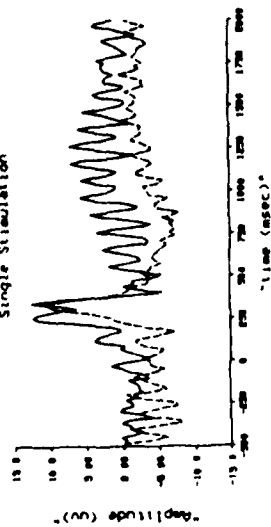


ELECTRODE: P4
SUBJECT: 5

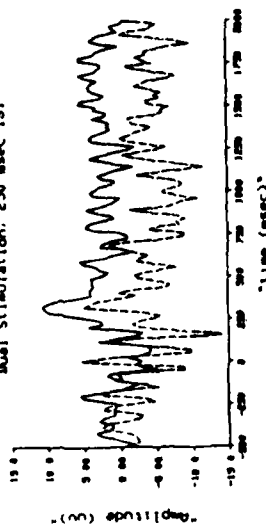
AVERAGES

2 CHOICE CASE - solid
4 CHOICE CASE - dash

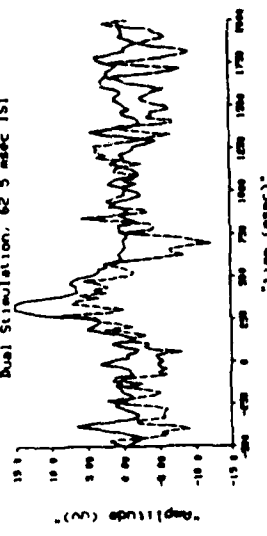
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

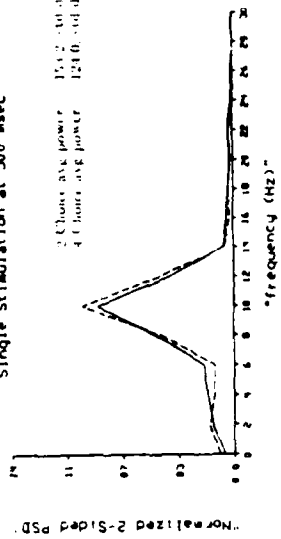


ELECTRODE: P4
SUBJECT: 5

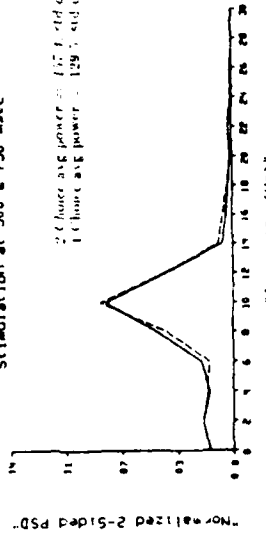
Normalized PSD
1/2 sec poststim

2 CHOICE - solid
4 CHOICE - dashed

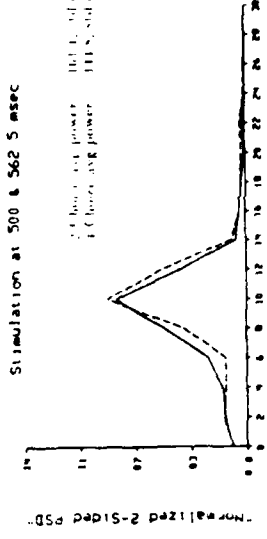
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



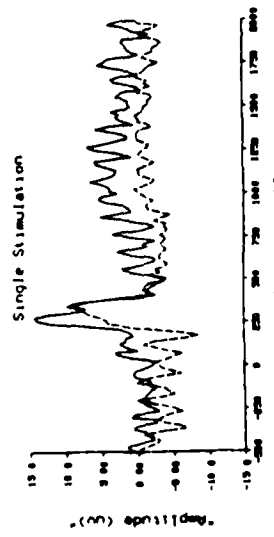
Stimulation at 500 & 562.5 msec



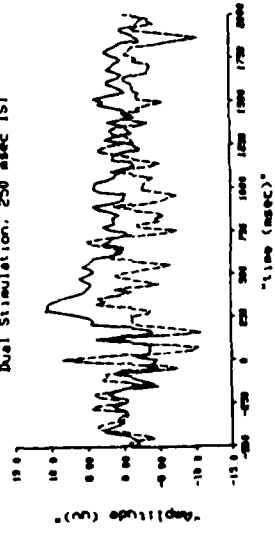
ELECTRODE: P3
SUBJECT: 1

AVERAGES

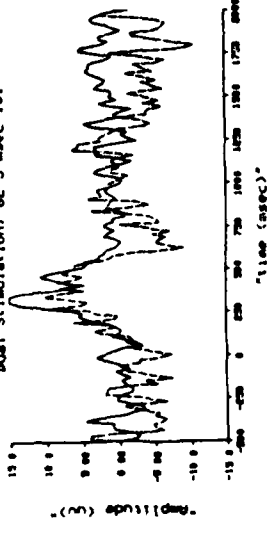
2 CHOICE CASE - solid
4 CHOICE CASE - dash



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

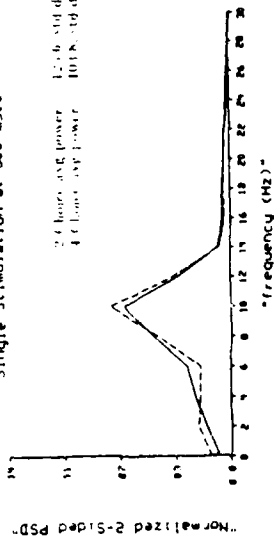


ELECTRODE: P3
SUBJECT: 1

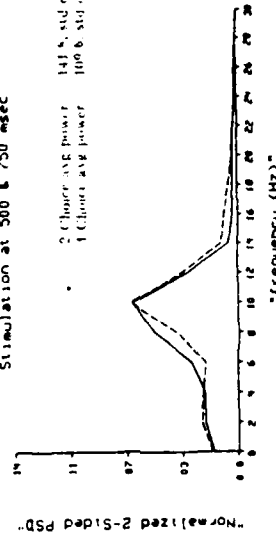
Normalized PSD
1/2 choice condition

2 CHOICE CASE - solid
4 CHOICE CASE - dashed

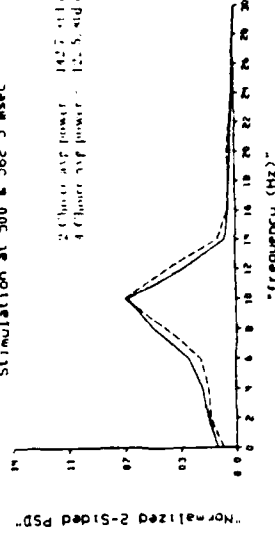
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



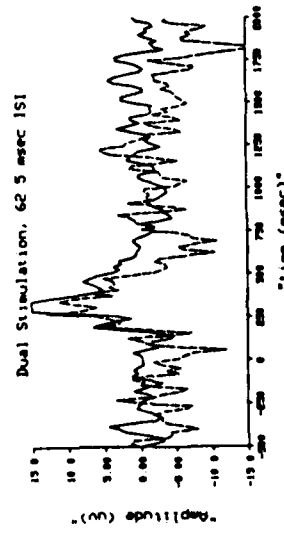
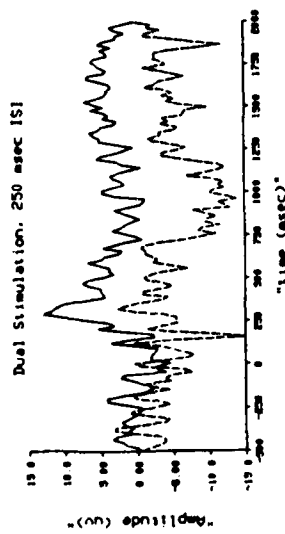
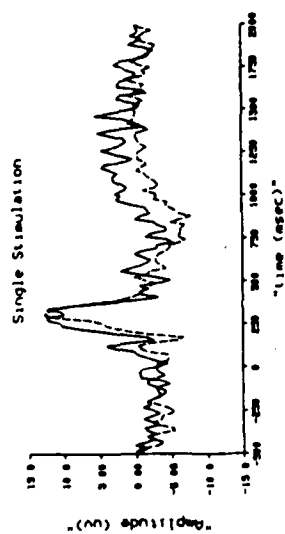
Stimulation at 500 & 562.5 msec



ELECTRODE C4
SUBJECT 5

AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

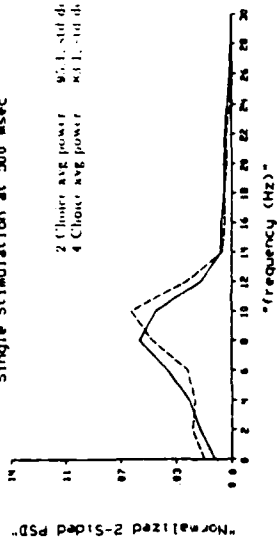


ELECTRODE C4
SUBJECT 5

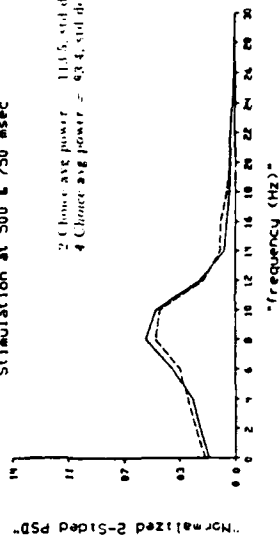
Normalized PSD
1/2 sec prestim

2 CHOICE CASE = solid
4 CHOICE CASE = dash

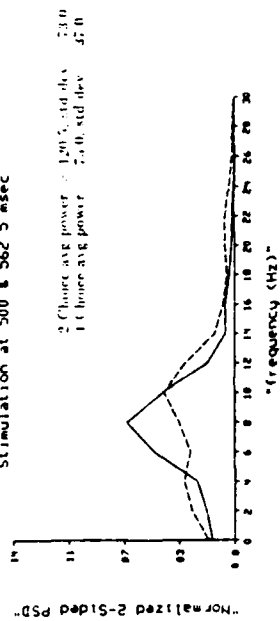
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec



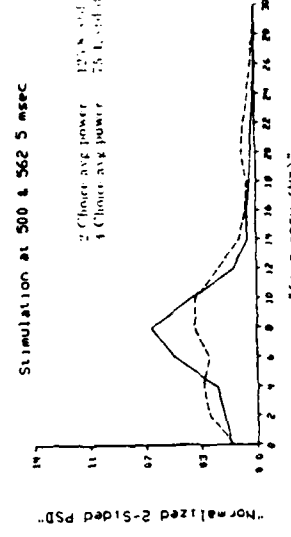
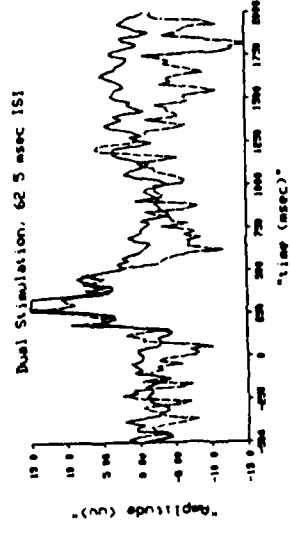
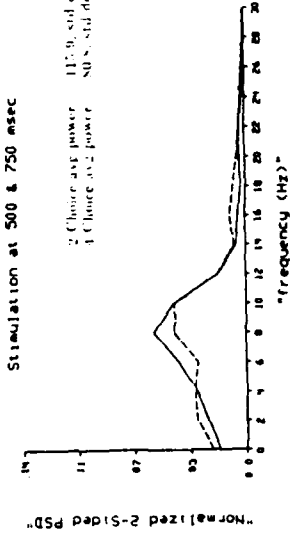
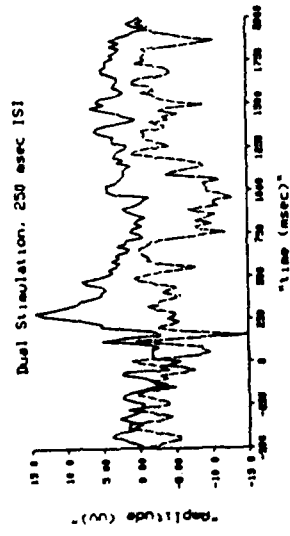
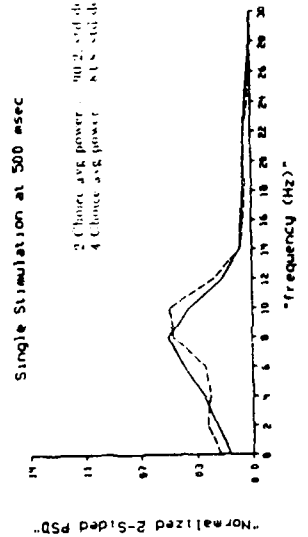
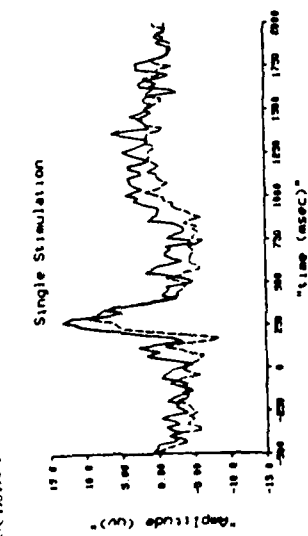
ELECTRODE C3
SUBJECT 5

AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dash



2 Choice avg power = 90.2 std dev = 18.1
4 Choice avg power = 81.8 std dev = 35.7

2 Choice avg power = 115.9 std dev = 35.0
4 Choice avg power = 80.8 std dev = 27.1

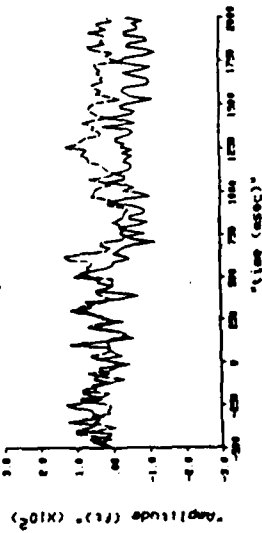
2 Choice avg power = 127.8 std dev = 37.9
4 Choice avg power = 75.1 std dev = 32.5

MEG, 7cm above right ear
SUBJECT 4

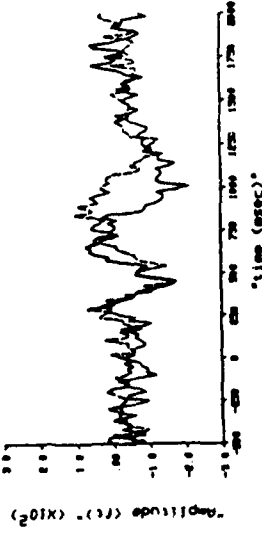
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

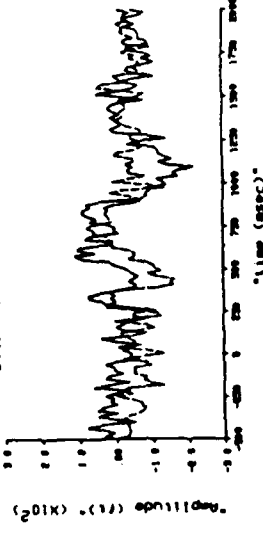
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI

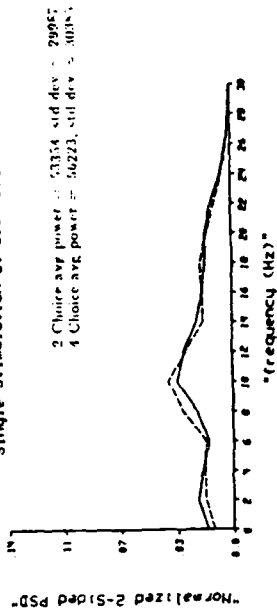


MEG
SUBJECT 4

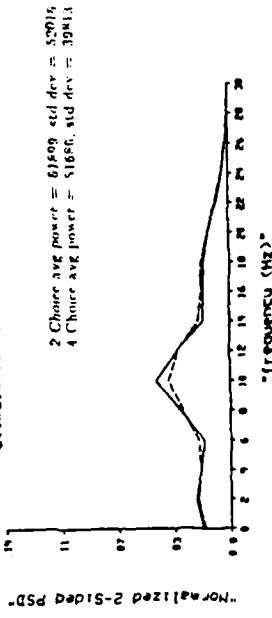
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

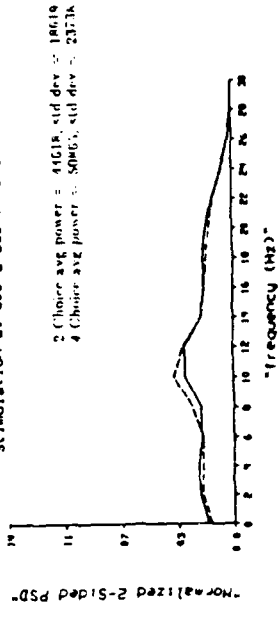
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



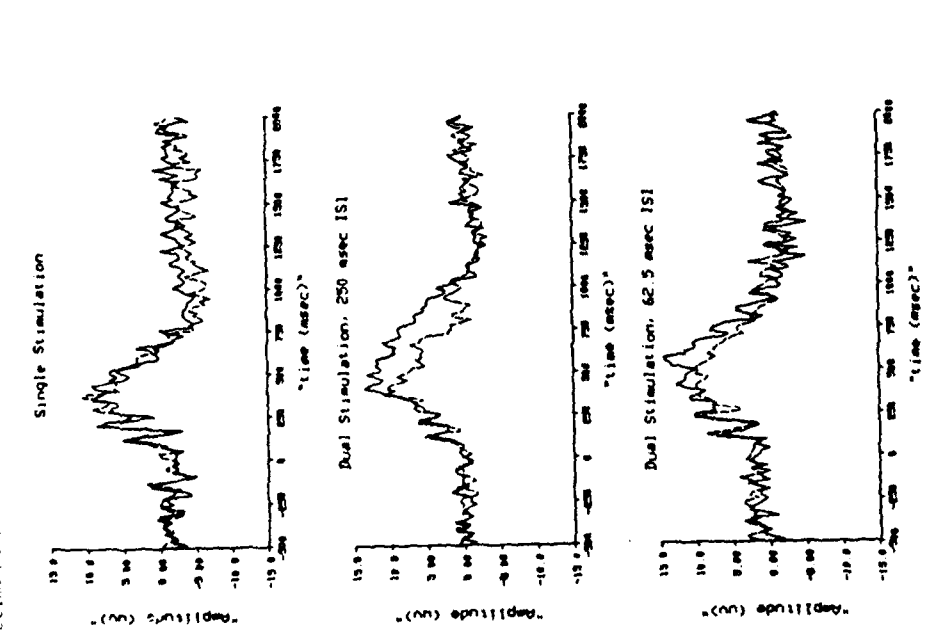
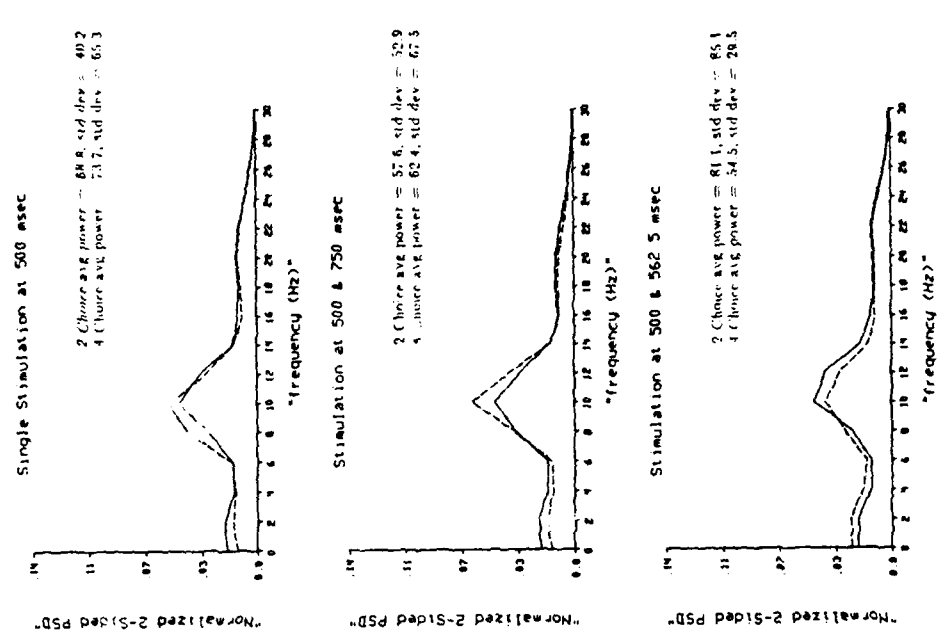
Stimulation at 500 & 562.5 msec



ELECTRODE: P4
SUBJECT: 4

AVERAGES
2 CHOICE CASE = solid
4 CHOICE CASE = dash

Normalized PSD
1/2 sec prestim
2 CHOICE = solid
4 CHOICE = dashed

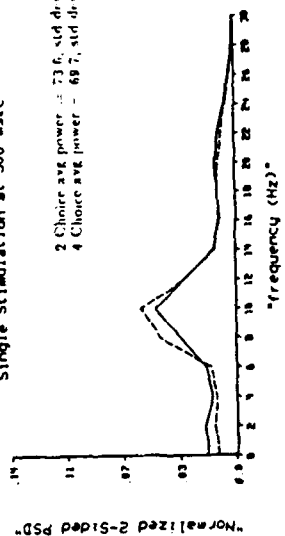


ELECTRODE P3
SUBJECT 4

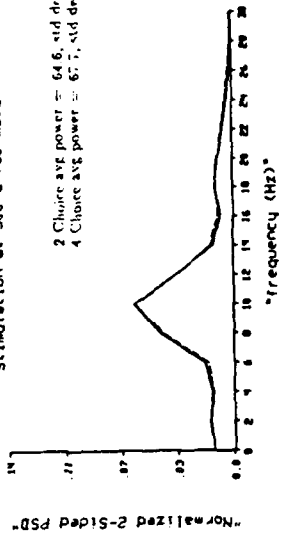
Normalized PSD
1/2 sec prestim

2 CHOICE = solid
4 CHOICE = dashed

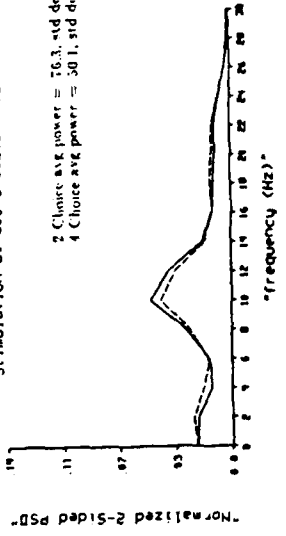
Single Stimulation at 500 msec



Stimulation at 500 & 750 msec



Stimulation at 500 & 562.5 msec

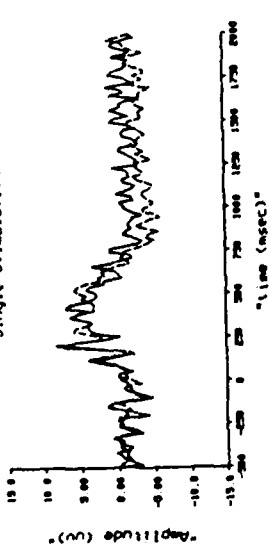


ELECTRODE P3
SUBJECT 4

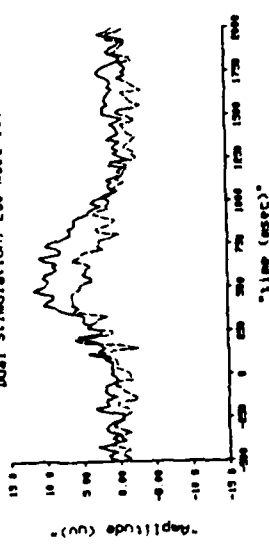
AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dash

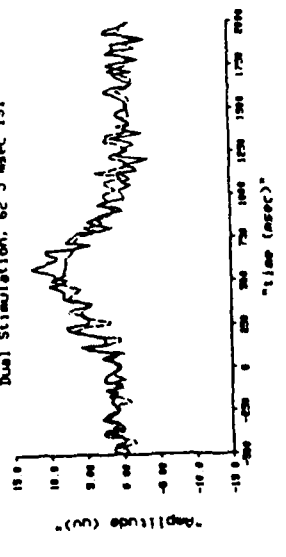
Single Stimulation



Dual Stimulation, 250 msec ISI



Dual Stimulation, 62.5 msec ISI



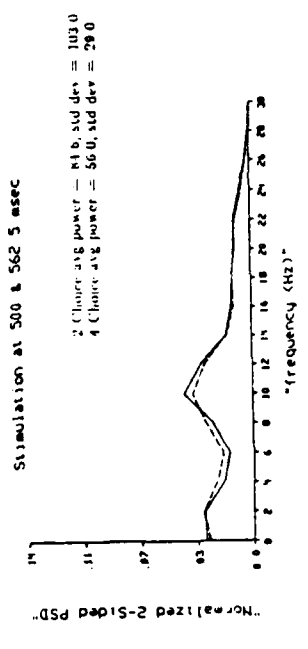
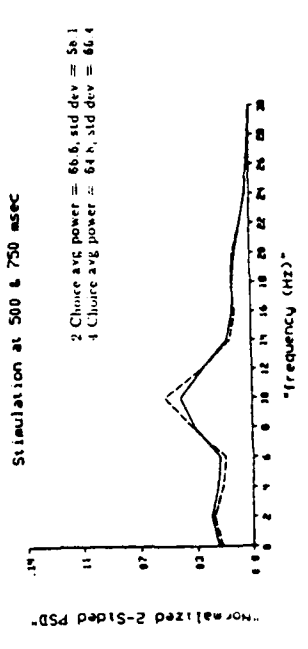
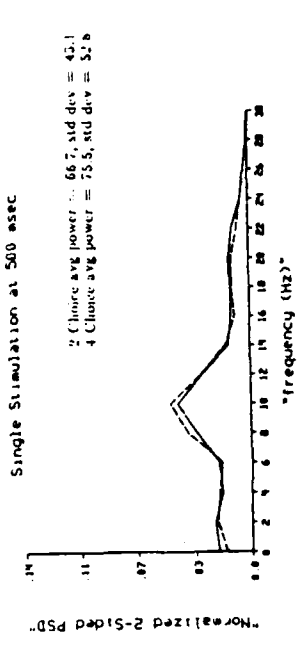
ELECTRODE C1
SUBJECT 1

AVERAGES

2 CHOICE CASE = solid
4 CHOICE CASE = dashed

Normalized PSD
1/2 sec prestim

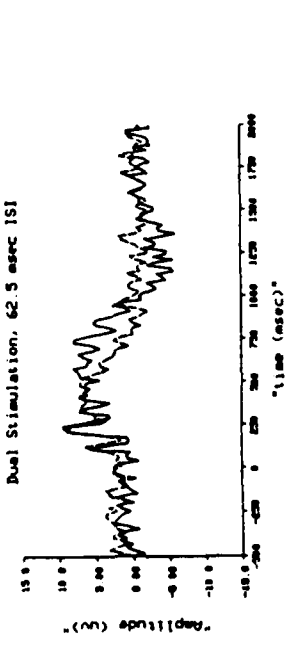
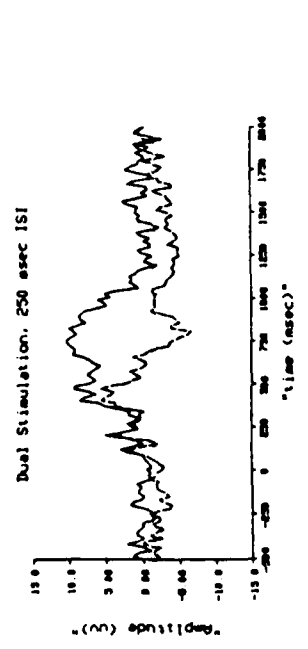
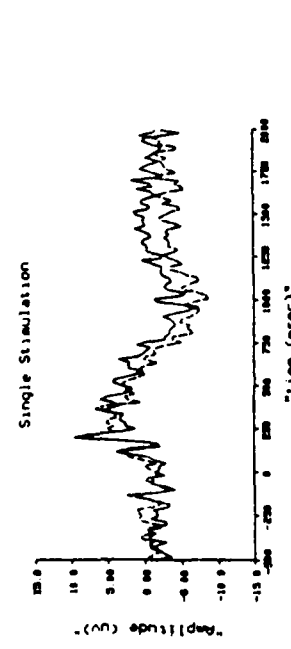
2 CHOICE = solid
4 CHOICE = dashed



ELECTRODE C4
SUBJECT 1

AVERAGES

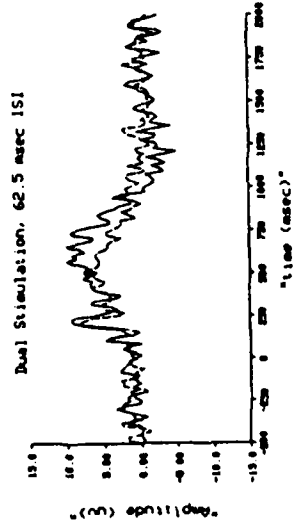
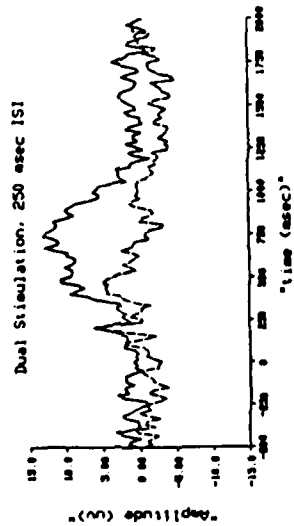
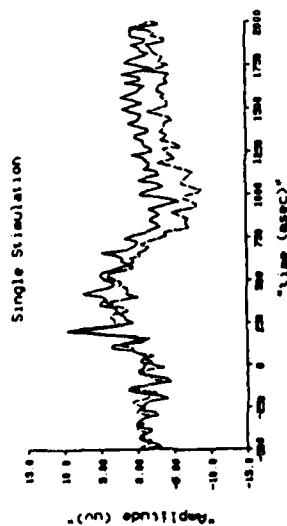
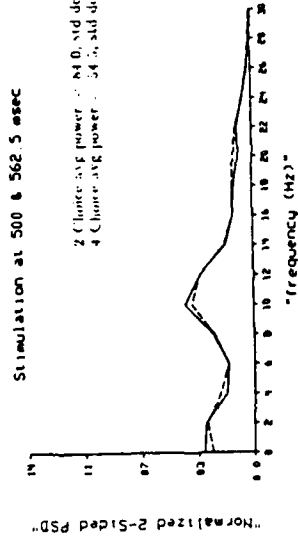
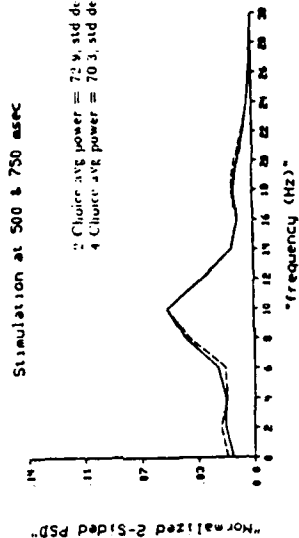
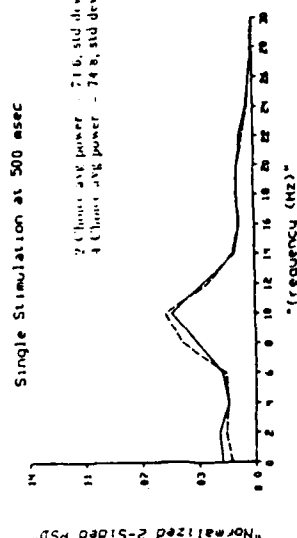
2 CHOICE CASE = solid
4 CHOICE CASE = dashed



ELECTRODE C3
SUBJECT 1

AVERAGES
2 CHOICE CASE = solid
4 CHOICE CASE = dash

Normalized PSD
1/2 sec prestim
2 CHOICE = solid
4 CHOICE = dashed



APPENDIX B

"MUTUAL" - SQUID Calibration Program

"MUTUAL" - SQUID Calibration Program

The mutual program is a short program that computes the mutual inductance and coupling between the SQUID detector coils and a test loop that is coaxial to them. It was originally written for and tested on a BTI (formerly S.H.E. Corp.) single channel DC SQUID with a second order gradiometer probe. Tables from Grover (1946, 1973) are used by the program to compute the mutual inductances. Once the mutual inductance has been computed, the coupling, or flux induced in the detector for a given current in the test loop is computed.

The program was written in the C programming language by Michael Plonski of Purdue University while on temporary assignment to the Naval Personnel Research Center (NPRDC) in San Diego. All measurements are relative to the bottom of the coil form on which the detector coils are wound. The bottom of the dewar tail is typically 0.394 inches from the bottom of the coil form. The detector coil dimensions are stored in define statements at the beginning of the program. These measurements should be verified with those supplied by the manufacturer. This program has also been used at the Aerospace Medical Research Lab at Wright Patterson Air Force Base where these values had to be changed to meet the specifications of their particular SQUID.

The source code for the program used at NPRDC is included here. A 1.2 inch radius test loop was made out of wire wrap wire and was used to test the program. This size loop fit snugly around the dewar tail and the remainder of the wire was twisted tightly with a drill to prevent any stray coupling from the wire to the detector loop. The test loop was moved from the bottom of the dewar tail (0.394 inches below the coil form bottom) to two inches above the dewar tail (1.6 inches above the coil form) and measurements were taken at 0.1 inch increments.

Figure B1 shows the comparison between the coupling predicted by the program and coupling measured by the SQUID. The measured coupling was simply the number of femtoteslas (fT) measured by the SQUID using the manufacturer supplied calibration, divided by the current in the loop as

measured with a oscilloscope and series resistor. All measurements were made using a 10 Hz sine wave and peak to peak values were compared. The predicted coupling was computed over a larger range from 0.5 inches below the coil form bottom to 2.5 inches above to demonstrate the symmetry as the loop passes over the various detector coils. The predicted and measured coupling closely overlay in the -0.4 to 1.6 inch range over which the measured values were taken. The peak coupling occurs when the test loop is centered over the four loops that comprise the middle set of the detector loops. Two nulls are present when the test loop is placed between the middle and lower set of coils and again when it is placed between the middle and upper set of coils. This is because the middle set of loops is wound in the opposite direction of the other loops.

A test fixture can be made by wrapping a coil around a wooden form that could be placed over the dewar tail. Care should be taken when designing a test fixture so that the test loop aligns over an portion of the curve with a low slope (preferably centered over the lower set of coils) in order to minimize errors due to placement. A second loop could also be placed over one of the nulls to check that the probe alignment has not shifted over time. The test fixture would also allow quick operational checks and ther ability to compute the transfer function for the entire measurement system rather than just the amplifier and filter stage.

"Inductance Calulations" 1946, 1973, F. Grover, published by the Instrument Society of America uner an arrangement with Dover Publications, Inc.

1.2 Inch Radius Calibration Coil

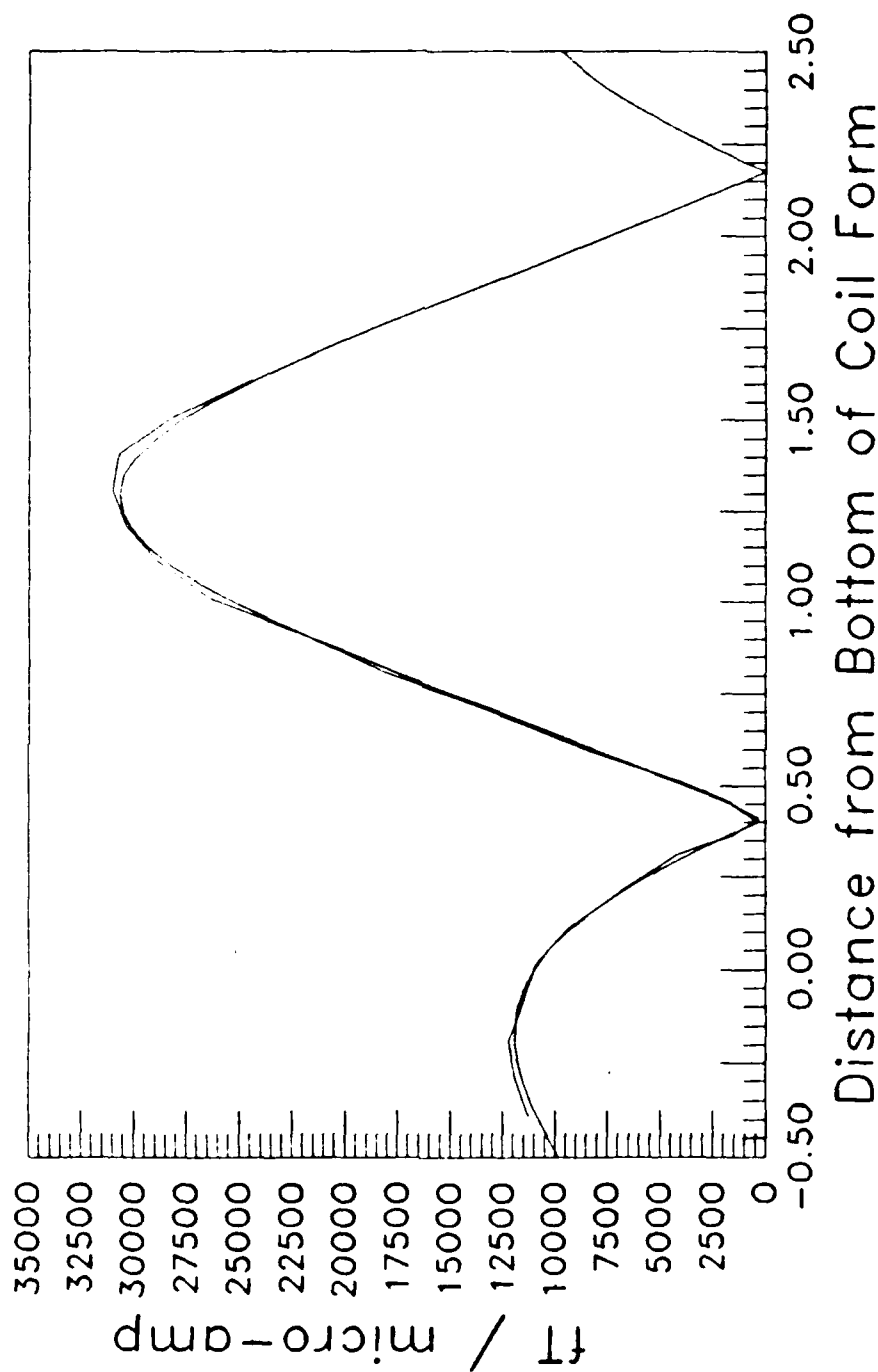


Figure B1 Predicted vs Measured Coupling

```

/*****
PROGRAM MUTUAL
*****/

```

THIS PROGRAM WILL COMPUTE THE EFFECTIVE MUTUAL INDUCTANCES BETWEEN A SINGLE LOOP AND THE SECOND ORDER GRADIOMETER DETECTION COILS USING EQS 77,78 & TABLE 13 OF "INDUCTANCE CALCULATIONS" BY GROVER, DOVER PUBLICATIONS. ALL VALUES ARE IN INCHES FROM BOTTOM OF QUARTZ COIL FORM. APPROXIMATE DISTANCE FROM BOTTOM OF COIL FORM TO BOTTOM OF DEWAR TAIL IS .394 in. PROGRAM WAS CHECKED USING DBX (C-debugger) TO ALTER THE VALUES TO THOSE USED IN EXAMPLE 25 OF GROVER.

Written by. Mike Plonski NPRDC, 6/85

```

*****/

```

```

#include <stdio.h>
#include <math.h>

/* FUNCTIONS */

extern float fval();
#define sq(x) ((x)*(x)) /* square function */
#define abs(x) ((x) > 0? (x): -(x)) /* absolute value func */
#define into:cm(x) ((x)*2.54) /* inch to cm convert */

/* DETECTOR COIL PARAMETERS & TABLE 13 */

float xl[] = {.027, .057}, /* location of lower coils */
xm[] = {1.247, 1.277, 1.307, 1.337}, /* middle coils */
xu[] = {2.527, 2.557}, /* upper coils */
rd = .4685, /* radius of det coils */
kinc = .01, /* increments in k for table 13 input */
f[] = { /* table 13 from grover */
    1.0, .021474, .017315, .014937, .013284,
    .012026, .011017, .010179, .009464, .008843,
    .008297, .007810, .007371, .006974, .006611,
    .006278, .005970, .005685, .005420, .005173,
    .004941, .004723, .004518, .004325, .004142,
    .003969, .003805, .003649, .003500, .003359,
    .003224, .003095, .002971, .002853, .002740,
    .0026317, .0025276, .0024276, .0023315, .0022391,
    .0021502, .0020646, .0019821, .0019026, .0018259,
    .0017519, .0016805, .0016116, .0015451, .0014808,
    .0014186, .0013585, .0013004, .0012443, .0011900,
    .0011374, .0010865, .0010373, .0009897, .0009436,
    .0008990, .0008558, .0008141, .0007736, .0007345,
    .0006966, .0006600, .0006246, .0005903, .0005571,
    .0005251, .0004941, .0004642, .0004353, .0004074,
    .0003805, .0003545, .0003295, .0003054, .0002823,
    .00025998, .00023859, .00021806, .00019840, .00017959,

```

```
.00016162, .00014450, .00012821, .00011276, .00009815,
.00008438, .00007146, .00005940, .00004824, .00003798,
.00002866, .00002035, .00001312, .00000708, .00000249,
0.0};
```

```
main()
{
```

```
/* INTERNAL VARIABLE DECLARATIONS */
```

```
float xc, /* location of cal coil relative to form bottom */
rc, /* radius of calibration coil */
sc, /* scale factor(cm)=sqrt(rc*rd) */
ml, /* mutual inductance with lower coil */
mm, /* middle coil */
mu, /* upper coil */
mt, /* effective mutual ind = mu - mm + ml */
coup, /* coupling coeff. in ft/micro amp */
pi= 3.14159265; /* standard definition */
```

```
/* GET CALIBRATION COIL PARAMETERS */
```

```
printf("Note Coil Form Bottom is .394 in from Bottom of Dewar Tail.0);
printf("Distance (inches) from Calibration Coil to Coil Form Bottom ? ");
scanf("%f",&xc);
printf("Radius (inches) of calibration coil ? ");
scanf("%f",&rc);
sc = intocm(sqrt(rc*rd));
```

```
/* COMPUTE MUTUAL INDUCTANCES */
```

```
ml = sc*(fval(rc,abs(xc-xl[0])) + fval(rc,abs(xc-xl[1]))); /* eq77 */
mu = sc*(fval(rc,abs(xc-xu[0])) + fval(rc,abs(xc-xu[1])));
mm = sc*(fval(rc,abs(xc-xm[0])) + fval(rc,abs(xc-xm[1])));
mm += sc*(fval(rc,abs(xc-xm[2])) + fval(rc,abs(xc-xm[3])));
mt = abs(mu - mm + ml);
```

```
/* PRINT OUT RESULTS */
```

```
printf(" MUTUAL INDUCTANCES (micro henrys) 0);
printf(" Lower Coil %f 0,ml);
printf(" Middle Coil %f 0,mm);
printf(" Upper Coil %f 0,mu);
printf(" Effective Total %f 0,mt);
```

```
/* COMPUTE AND OUTPUT COUPLING COEFFICIENT */
```

```
/* coupling = mutual inductance / effective area, for compatibilty with */
/* S.H.E. field sensitivity value use combined area of bottom two turns */
```

```
coup = 10000000.0 * mt / (2*pi*sq(intocm(rd)));
printf(" Coupling Coefficient = %8.1f femto-tesla / micro-amp 0,coup);
```



```

    exit(0);
}

float fval(rc,d) /* return interpolated value from table 13 */
float rc,        /* radius of calibration coil */
d;              /* distance between cal coil & current det coil */
{
    float k,                /* input parameter for table 13 */
          kdelta,          /* norm diff between true k & table entry used */
          d1,              /* first difference in table */
          d2,              /* second difference in table */
          value;           /* final value to return */
    int ik;                /* floor index for k in table */

    k = (sq(rd-rc) + sq(d))/(sq(rd+rc) + sq(d)); /* eq. 78 */
    ik = k/kinc;
    kdelta = (k - ik*kinc)/kinc;
    if(k < .1 || k > .9) /*** WARNING **/
    {
        printf("warning: k'2 = %f is outside desired range for this table0,k);
        printf("      see Grover (ch. 11) for alternate tables 0);
    }
    if(ik == 0 || ik > 97) /*** ERROR **/
    {
        printf("error: k'2 = %f is out of range for this table 0,k);
        exit(1);
    }
    d1 = f[ik+1] - f[ik];
    d2 = f[ik+2] - 2.0*f[ik+1] + f[ik];
    value = f[ik] + kdelta*(d1 - (1.0 - kdelta)*d2/2.0);
    return(value);
}

```

APPENDIX C

Data Acquisition Code

DATA ACQUISITION CODE MAINTAINACE GUIDE

OVERVIEW:

The data acquisition package was designed to be very flexible in allowing the Tecmar Labmaster board and IBM PC/AT to collect data and control experiments. The basic system is shown in Figure C1. The package is capable of presenting and recording 8 bits of stimulus and response data while timing two sets of external events and digitizing eight channels of data. Warning tones can also be presented to alert the user to prepare for a stimulus. All of the data collection is interrupt driven which allows the user to perform real time analysis of the data while it is being collected. The A/D channels are double buffered so that the previous trial can be analyzed while the data for the current trial is being collected. The program is controlled by the input parameter file which may be prepared in advance and modified at run time. The makepar program is a program that will generate a parameter file based on prompting the user for inputs. The data collection of the D/A channels is broken down into trials and the user can have as many trials as disk space will allow. The external interrupt lines are monitored continuously during the course of the experiment and are independent of the trials. Each trial can present two independent 8 bit stimuli and record two 8 bit responses. The responses interrupt handler can be used to turn the stimuli off. Up to 64K bytes of digitized data can be recorded per trial.

The timing for an individual trial is shown in Figure C2. The delay time (in msec), foreperiod (in msec), ISI (in sample points) and warning tone frequency can be varied on a trial by trial basis. Both stimuli can also be varied on a trial by trial basis. The warning tone duration, sampling length and prestimulus interval (for the first stimulus) are fixed at the start of each experiment. All of these variables, as well as the sampling rate, number of responses and number of external interrupt lines are controlled by the parameter file.

TECMAR CONTROL USING MNEMONIC CODES:

Much of the mystery of programming the tecmar lab-master card has been alleviated through the use of mnemonic definitions contained in the tecmar.h and param.asm files. The tecmar.h file is not an exhaustive list of command codes, however it's scope is more than adequate to handle all of the command codes necessary for the current software. The param.asm file contains a subset of the definitions that are in the tecmar.h file along with general definitions for the assembly language routines. General definitions for the C language routines are in the param.h file.

The first section of tecmar.h contains the address of all the internal tecmar registers along with their mnemonic equivalent. The remainder of the file consist of general command codes as well as application specific codes for these registers. An example of a general mnemonic definition from tecmar.h is:

```
#define LOADTX 0x08 /* T_CONT = LOADTX | TX; T_DATA -> x load reg */
```

The convention of using capital letters for C constants is followed and also used for assembly language constants. The 0x prefix is the standard prefix to indicate a hexadecimal number. Functionally, this statement only sets the constant LOADTX equal to 8. Organizationally, this allows us to write mnemonic rather than cryptic code. The comment to the right of the definition explains its use. When the timer control register (T_CONT) is set to the command code for load register selection (LOADTX) is or'ed with the register number (T1, T2, T3, T4 or T5), the timer data register will address the specified load register (1, 2, 3, 4, 5 respectively). An example of the use of this definition is:

```
outh(T_CONT,LOADTX | T5); /* point timer data reg. to timer 5 load reg */  
out2b(T_DATA, INIT_DELAY); /* set timer 5 load reg to the initial delay */
```

T5 is just a constant equal to 5. This may seem like an excessive use of mnemonics, however some command codes identify timers by bit number. The command code to arm and load timer 5 (A_LSX | S5) is one such example. S5 is equal to 0x10 (bit 5 set), while T5 is equal to 0x05 (the number 5) yet both are used to identify timer 5, but to different command codes.

Mnemonic codes are used for all references to the tecmar board, with the exception of the number 0, to enhance the readability of the code. While it may be possible to program the tecmar board solely through the use of these mnemonics, nothing can replace an understanding of how the board actually functions. The command "outb(P_CONT, PA_2M);" which configures port A for mode 2 operation does not tell you very much if you do not know what mode 2 is. The mnemonic codes can also be a valuable asset while reading the tecmar manual as they can help clarify some of commands. NOTE that approximately 20% of the mnemonics have not been used in the current software and are therefore untested. One may wish to verify the formulation of specific mnemonics before using them for other applications.

MEMORY ALLOCATION:

The IBM-PC operating under the DOS 3.x operating system is a 16 bit machine and consequentially a data segment contains 64K bytes of storage. The small model C compilers assume a single segment for data storage and therefore use 16 bit pointers. This is not adequate for our needs, so that the large model compilers must be used which allow for multiple data segments through the use of 32 bit pointers. It is imperative to remember that all pointers contain both a segment and offset value when interfacing to assembly language routines.

There are four types of data structures which are used in the program and defined in the param.h. The param.asm file contains an equivalent definition of the IC_STRUCT data, since this is the only structure used by the assembly language code.

- 1) INT_STRUCT interrupt structure which is used to store the original ibm interrupt vectors before revectoring by the dac program
- 2) RESP_STRUCT response structure used to store the stimuli, responses and responses time
- 3) PAR_STRUCT parameter structure which is used to hold all of the parameters which specify the program operation

- 4) IC_STRUCT interrupt control structure is used to pass information between the high level C routines and the low level assembly interrupt handlers. It is crucial that the C structure definition in param.h match the assembly definition in param.asm EXACTLY.

The dac.c file allocates storage for the program parameters, responses and interrupt revectoring using the first three data structures. The storage.asm file allocates several segments for storage, including the dataseg-ment which contains the interrupt control structure. Two 64K segments (buffer0 and buffer1) are allocated to store the data from the A/D conversions in 2 byte interger form. On odd trial numbers, data is written to buffer1, while on even trials it is written to buffer0. This double buffering allows us to write out the results from trial n-1 while data is collected for trial n. Two smaller segments (external0 and external1) are allocated to provide storage for the timing of external events in 4 byte long interger form. Storage is allocated in the storage.asm file.

INTERRUPT HANDLING:

There are three types of interrupts (timer, response, and external) that may occur and each type has its own interrupt handler. The interrupt request lines are defined in the param.h file and the comments in the tec_init routine show how the board should be jumpered for the current configuration. The tec_init routine initilizes all of the tecmar control registers and uses the set_int routine to alter the normal IBM interrupt mask and vectors. Any premature termination of the program which takes place after the tec_init routine, MUST call the error routine to reset the IBM-PC to it's original state. Failure to restore the interrupt masking & vectoring will result in ERRATIC MACHINE BEHAVIOR. This can also occur if one "breaks" out of the program. The IBM-PC can be rebooted to restore the IBM-PC to it's correct state. A real time 32 bit clock (timers 3&4) running at 10 KHz is started at the beginning of the experiment and is used by all of the interrupt routines to record the time of events (e.g. stimulation, response, external interrupts). The clock has an accuracy of 0.1 milliseconds over a period that can be as long as 119 hours.

TIMER INTERRUPT:

The `tm_int` routine is the interrupt handler for timer 5. Timer 5 is used for both delay and sample timing and is operated in a count down mode which generates an interrupt when the count reaches zero. If the sampling bit (`F_AD`) in the flag byte of the interrupt control structure (`ic.flags`) is set, then a set of samples from the A/D converters is taken (200 microseconds to sample 6 channels). The routine then examines the sample number to determine if a stimulus should be turned on or if this was the last set of samples to take for this trial. Note that the second stimulus sent to the output port (Port B) is equal to the current stimulus in the port register exclusively or'ed with the second stimulus stored in the parameter structure. This allows the stimuli to be set and reset (by the response routine) independently of each other; or to use the second stimulus to reset the first stimulus by using the same bits for both. The time of the first stimulus is read from the real time clock and stored to facilitate determination of response times and data alignment with externally timed events. The time of the second stimulus can be deduced from the number of sample points between the stimuli. If the sampling bit is not set, then the routine assumes that only a delay was requested and sets the delay finished flag (`F_DELAY`). Since no sampling is involved, this type of interrupt can be handled in approximately 30 microseconds. The interrupt line for a delay is generated from a 1 KHz clock, so that this line is forced low within the interrupt routine to prevent the same clock cycle from causing multiple interrupts.

RESPONSE INTERRUPT:

The `res_int` routine is used to record both the response time and response byte. It will record up to two responses per trial and the interrupt from the second response is ignored if the second response is the same as the first response. This means that if you press the first response five times and then press the second (different) response, only the first time that the first response was pressed and the time of the second response will be accepted and stored. The current response byte is compared to the appropriate stimulus byte, if they match the current stimulus is and'ed with the inverse of the current response. Note that this is different from simply clearing the stimulus byte and allows independent operation of the two stimuli and responses. Flags (`ic.flags` byte, bits `F_RESP0` & `F_RESP1`) are used to indicate which response byte is active and interrupts received with no active

response flag byte are ignored. Both response flags are set at the start of sampling so that responses given during the prestimulus interval are recorded with a negative response time. This allows us to check for anticipation on the part of the subject. When we write the modified stimulus to Port B (output port), we also clear the interrupt line associated with B. This can pose a serious problem, since the Port B interrupt line is used to indicate an external event on one of the two data bits assigned to Port C. This situation is discussed further in the external interrupt handler section and the hardware section which describes the hardware solution to this problem. The hardware is configured so that any button push will latch the switch configuration into Port A (input port) and generate an interrupt. Switches were also made bounceless by using an RC charge and discharge network in combination with a single pole single throw switch. The response is read from the Port A register and the time is read from the real time clock.

EXTERNAL INTERRUPTS:

The `ext_int` routine is the interrupt handler that is used to respond to external interrupts occurring on the two data bits assigned to Port C (bits C6 & C9) and generated using the Port B interrupt request line (INTRb). Port B is currently configured as an output port and drives a series of LEDs. Port C is currently used to threshold detect the EKG and respiratory waveform. The interrupt request line on Port B in this configuration was designed to function as a ready to receive signal. Since we are only driving LEDs we do not need this function and instead use the Port B interrupt to indicate that an external event has occurred on one of the two data bits of Port C. Unfortunately, every time that we write to Port B we clear the interrupt request line. A hardware solution (see hardware section) was constructed based on generating a second interrupt on Port B if the first interrupt line was cleared in less time than it took for the `ext_int` routine (approximately 30 microseconds) to handle the interrupt. This solution is satisfactory proved that the Port B interrupt request line priority is at a higher level than the other interrupt priorities. Port C only has two data bits available since the other six bits are used for handshaking on Ports A and B. The interrupt handler simply reads in Port C and determines which external event occurred (bit C6 or C7 set) to determine which data segment to store the time of the event in.

HARDWARE:

An interface box, shown in Figure C1, was constructed to connect the Tecmar data acquisition to the outside world. The input port (Port A) and the output port (Port B) were both buffered in this box to prevent damage to the Tecmar board if an improper connection was made. The main circuitry in the box was the interrupt arbitrator circuitry shown in Figure C3. This circuit served two purposes. The circuit had to be able to correctly handle simultaneous interrupts on both external interrupt input lines. It also had to address the problem of using the Port B interrupt line for data on Port C. This problem was addressed in was discussed in the external interrupt section. The term simultaneous can be defined as two events occurring within the time required to service a single interrupt (approximately 30 microseconds). This is a more than adequate definition of simultaneity since the real time clock only measured time in increments of 100 microseconds.

The resistor capacitor combination at the input of the circuit is used to set the threshold level and provide some low pass filtering of the input lines. Note that in this configuration the respiration channel was triggered by the start of an expiration and the EKG channel was triggered by the rising edge of the R wave if the input polarities were set properly. The Schmitt trigger inverter (U9) is used to prevent multiple threshold crossings from signals with a slow slew rate. The multivibrator (U7) and flip flop (U4) were used to handle the problem of pseudo simultaneous threshold crossings. The line which crosses the threshold second will generate a second interrupt as soon as the system finishes with the first interrupt. The operation of this circuit was tested and verified with pulses that were separated by 1 microsecond. The next set of multivibrators (U8) are used to reinterrupt the system if the interrupt line was cleared prematurely by one of the other routines writing to Port B. This circuit was successfully used on 10 different subjects and the only problems encountered were with partial breaths which did not cross the threshold level.

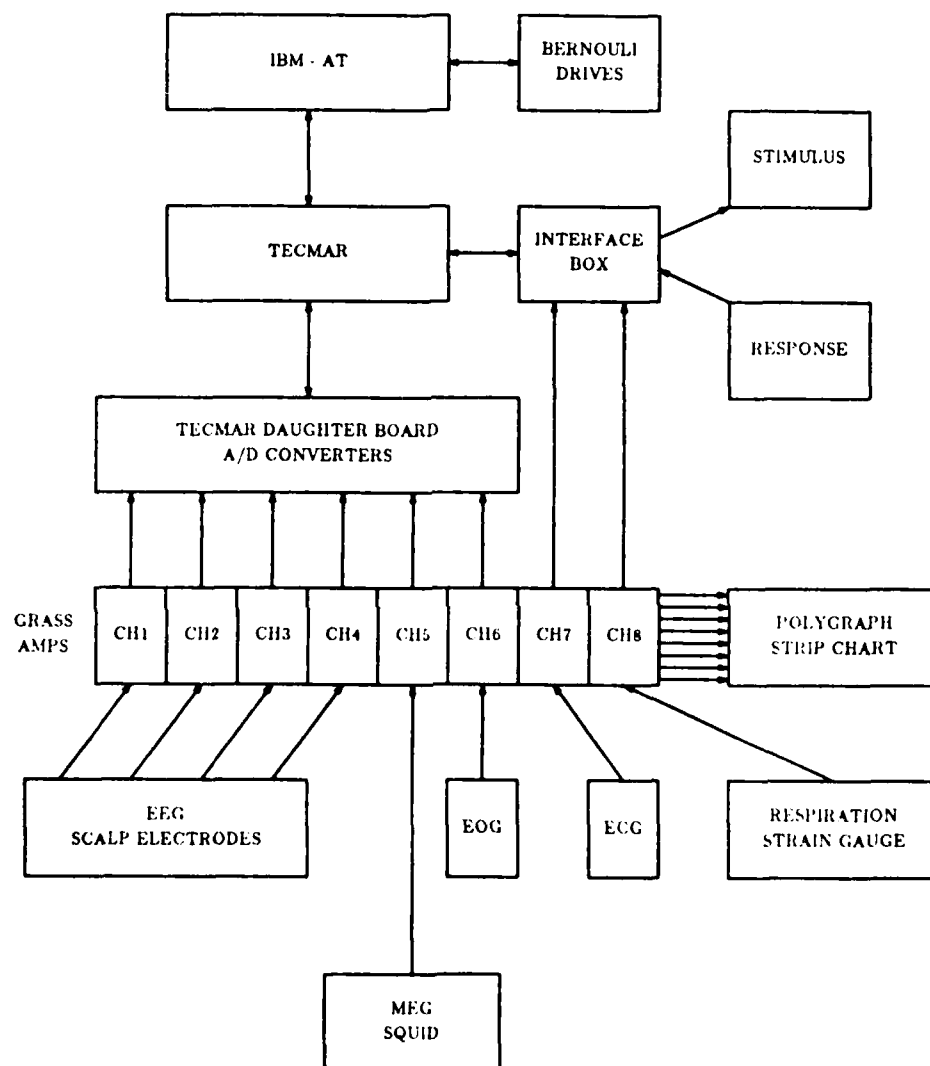


Figure C1 Equipment Configuration

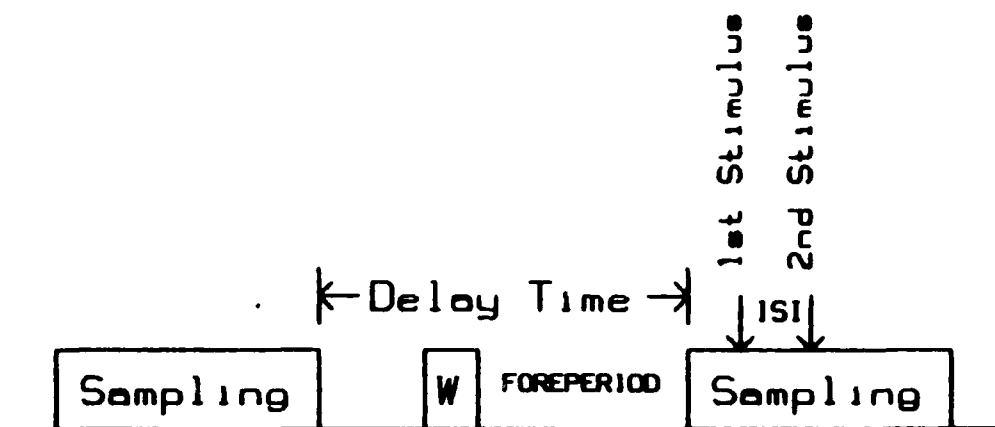


Figure C2 Single Trial Timing Diagram

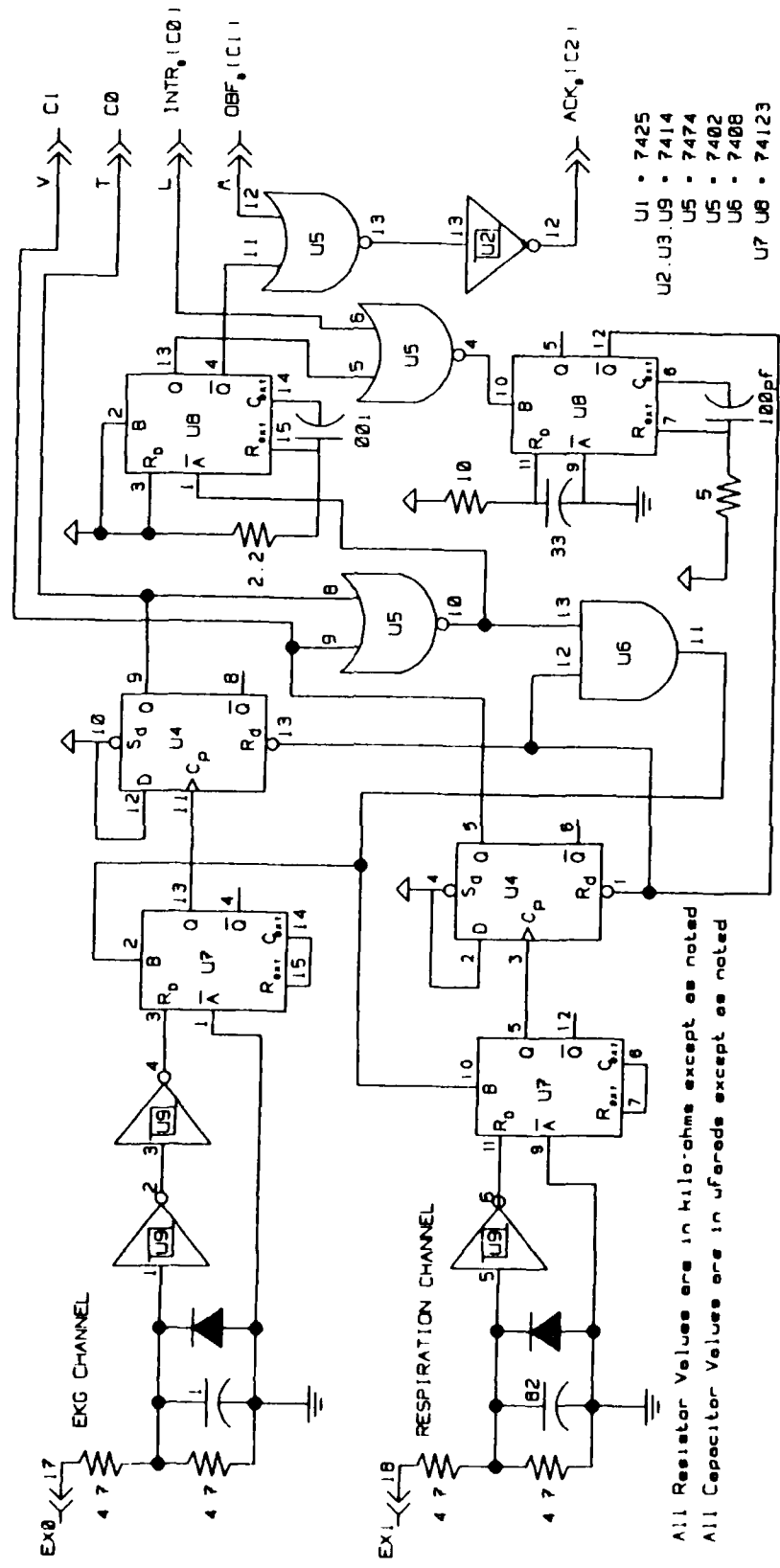


Figure C3 Interrupt Arbitration Circuit

```
.....
```

SUBROUTINE - CHECKPAR(fp)

```
.....
```

PURPOSE:

check the parameter structure & allow user to review

INPUT:

external - par: parameter structure defined in param.h

OUTPUT:

external - par (modified)

DEPENDENCIES:

getline() - function used to read in 1 line at a time

stdio.h

math.h

param.h

```
...../
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include "param.h"
```

```
#define getp(form,v) if(getline(s) != 0) sscanf(s,form,v)
```

```
void checkpar()
```

```
{
extern PAR_STRUCT par;           /* parameter structure */
int c0, c1;                      /* temp char to make stimulus word */
char s[MAXLINE];                 /* string for inputting parameters */
int i;                           /* temporary index */
    min_delay, max_delay,        /* min & max ISI delays */
    min_wtper, max_wtper,        /* min & max of warning tone periods */
    min_wtfore, max_wtfore,      /* min & max of warn tone foreperiod */
    min_isi, max_isi;            /* min & max of #samps between stims */
```

```
printf("Do you wish to review the parameters selected ?");
```

```
getline(s);
```

```
while(s[0] == 'Y' || s[0] == 'y'){
```

```
    printf("Enter new value or CR to keep old 0);
```

```
    printf("number of channels -- %d ?",par.nchan);
```

```
    getp("%d",&par.nchan);
```

```
    printf("number of responses expected (0,1 or 2) = %d ?",par.response);
```

```

getp("%d",&par.response);

printf("number of external interrupt lines (0,1 or 2) = %d ?",par.external);
getp("%d",&par.external);

printf("number of trials (max = %d) = %d ?",MAX_TRIAL,par.ntrial);
getp("%d",&par.ntrial);

printf("sampling period in usec (max = 64000) = %d ?",par.s_per);
getp("%d",&par.s_per);

printf("warning tone duration in msec = %d ?",par.wt_dur);
getp("%d",&par.wt_dur);

printf("number of samples per trial = %d ?",par.s_total);
getp("%d",&par.s_total);

printf("sample # after which to turn stimulus 0 on = %d ?",par.s_stim0);
getp("%d",&par.s_stim0);

min_delay = max_delay = par.delays[0]; /* initialize min & max */
min_wtper = max_wtper = par.wt_pers[0];
min_wtfore = max_wtfore = par.wt_fore[0];
min_isi = max_isi = par.s_isi[0];
for (i=0; i < par.ntrial; i++){
    min_delay = min(min_delay, par.delays[i]);
    max_delay = max(max_delay, par.delays[i]);
    min_wtper = min(min_wtper, par.wt_pers[i]);
    max_wtper = max(max_wtper, par.wt_pers[i]);
    min_wtfore = min(min_wtfore, par.wt_fore[i]);
    max_wtfore = max(max_wtfore, par.wt_fore[i]);
    min_isi = min(min_isi, par.s_isi[i]);
    max_isi = max(max_isi, par.s_isi[i]);
}

printf("1st trial stimulus = %x %x & last trial stimulus = %x %x0,
       par.stims[0] & 0x00ff, (par.stims[0] > 8) & 0x00ff, par.stims[par.ntrial
       -1] & 0x00ff, (par.stims[par.ntrial-1] > 8) & 0x00ff);
printf("ISI (in # samples) range from %d to %d 0,min_isi,max_isi);
printf("inter trial delay range from %d to %d msec0,min_delay,max_delay);
printf("warning tone periods go from %d to %d msec0,min_wtper,max_wtper);
printf("WT fore periods range from %d to %d msec0,min_wtfore,max_wtfore);
printf("Do you wish to review all %d trials ?",par.ntrial);
getline(s);
if(s[0] == 'y' || s[0] == 'Y'){
    printf("0im(hex pair), ISI(#samps), trial delay(ms), WT period(us), WT foreperiod(ms))0);
    for(i=0; i < par.ntrial; i++){
        printf("(%x %x, %d, %d, %d, %d) ?",par.stims[i] &
            0x00ff, (par.stims[i] > 8) & 0x00ff, par.s_isi[i], par.delays[i],
            par.wt_pers[i], par.wt_fore[i]);
        if(getline(s) != 0){
            sscanf(s, "%x %x %d %d %d %d", &c0, &c1, &par.s_isi[i],
                &par.delays[i], &par.wt_pers[i]);

```

```

        par.stims[i] = c0 + (( c1 <= 8) & ~0xFF);
    }
}
printf(" Oview Parameters Again ?");
getline(s);                                /* return to top of loop */
}

/*    CHECK RANGE OF PARAMETERS */

if(par.response > 2) error("mkpar: par.response > 2");
if(par.external > 2) error("checkpar: par.external > 2");
if(par.ntrial > MAX_TRIAL) error("checkpar: par.ntrial > MAX TRIAL ");
for(i=0; i< par.ntrial; i++){
    if(par.delays[i] < par.wt_dur + par.wt_fore[i])
        error("checkpar: par.delays < par.wt_dur + par.wt_foreperiod");
    if(par.s_isi[i] == 0)
        error("checkpar: par.s_isi == 0");
    if(par.s_isi[i] + par.s_stim0 == - par.s_total)
        error("checkpar: par.s_st01 + par.s_stim0 == - par.s_total");
}

return;
}

```

```
.....
```

PROGRAM - DAC

```
.....
```

PURPOSE:

data acquisition program

```
...../
```

```
#include <stdio.h>
#include "tecmar.h"
#include "param.h"
```

```
PAR STRUCT par;          /* allocate space for parameter structure */
RESP STRUCT resp[MAX_TRIAL]; /* allocate space for responses & times */
INT STRUCT tm,           /* allocate space for timer int vector */
        res,             /* allocate space for response int vectors */
        ext;             /* allocate space for external int vectors */
```

main()

```
{
    setup();
    tec init();
    while (samp()){
        /* enter task to be done while sampling here */
        delay();
        /* enter task to be done during delay here */
    }
    tec done();          /* reset & disable tecmar & int */
    store();             /* store & close all files */
    exit(0);             /* normal program exit */
}
```


/*****

FUNCTION SUBROUTINE - DELAY()

*****/

PURPOSE:

Perform all necessary operations after sampling is completed for a trial.

- 1) sampling done flag set -> call error & terminate
sampling finished before high level processing == fatal error
- 2) wait for sampling done flag to be set
- 3) reset sampling done flag
- 5) check if this was last trial
- 7) start delay in background

INPUT:

none - external ic, par

OUTPUT:

none

RETURN (int):

- 1 = normal sampling - inter trial delay & warning tones initiated
- 0 = normal finish - no delay initiated
- use startsamp to store last trial

DEPENDENCIES:

tm int() - performs actual delay in background
error() - error & terminate routine
ic - interrupt delay structure (extern)
par - experiment parameter structure (extern)
tecmr.h - parameters for tecmar board
param.h - general parameters
stdio.h - C standard input/output

*****/

```
#include <stdio.h>
#include "param.h"
#include "tecmr.h"
```

```
int delay()
```

```
{
extern PAR_STRUCT par;          /* experiment parameter structure */
extern IC_STRUCT ic;           /* interrupt control structure */
int temp;
```

```
/* CHECK SAMPLING DONE FLAG */
```

```

    if(ic.flags & F_SAMP) error("endsamp: sampling finished early");

/* WAIT FOR SAMPLING FLAG & THEN RESET IT */

    while(!{ic.flags & F_SAMP});
    ic.flags &= ~(F_SAMP | F_AD | F_RESP0 | F_RESP1);
                    /* clear samp done, a/d, waiting for response flags */

/* CHECK IF FINISHED WITH TRIALS - RETURN (0) - last trial not stored yet */

    if (++par.itrial == par.ntrial) return(0);

/* START DELAY & WARNING TONES */

    printf("starting delay for trial # %d 0.par.itrial);
    outb(T_CONT,T5);
    out2b(T_DATA,TM5ms);          /* set timer 5 for msec delays */
    out2b(T_DATA,par.delays[par.itrial]); /* set delay in load reg */
    out(T_CONT,A_LSX, S5);        /* start delay */

    if (par.wt_dur != 0) {          /* produce warning tones */
        outb(T_CONT,LOADTX | T1); /* T_DATA -> timer 1 load reg */
        temp = par.delays[par.itrial] - par.wt_fore[par.itrial] - par.wt_dur +
            (par.s_per * par.s_stim0) / 1000 ;
        out2b(T_DATA, temp);
        outb(T_CONT,LOADTX | T2); /* T_DATA -> timer 2 load reg */
        out2b(T_DATA, par.wt_pers[par.itrial]);
        outb(T_CONT, A_LSX | S1 | S2); /* start warning tone generator */
        if(!HIGH_FREQ || F1 / (long)par.wt_pers[par.itrial])
            outb(HB_0DA,0x08); /* output 0 volts (0 - 10 range) */
            outb(LB_0DA,0x00);
        else {
            outb(HB_0DA,0x00); /* output 5volts (0 -10 range) */
            outb(LB_0DA, 0x00);
        }
    }

/* RETURN NORMALLY */

    return(1);

```

/******

SUBROUTINE - ERROR(msg)

PURPOSE:

print error message & terminate program

INPUT:

msg - pointer to error message to be printed out

OUTPUT:

none - execution terminates

DEPENDENCIES:

exit() -- standard C exit routine
tecmar.h -- tecmar constant definitions
param.h -- general parameter & structure definitions
stdio.h -- C standard input/output

*****/

```
#include <stdio.h>
#include "param.h"
#include "tecmar.h"
```

```
void error(msg)
```

```
char *msg;                /* error message to be printed out */
{
    extern PAR_STRUCT par;
    outb(T_CONT, DISAUTO); /* disable timer interrupts */
    outb(P_CONT, PC_RSET | INTE_AIN); /* disable PORTA interrupts */
    outb(P_CONT, PC_RSET | INTE_B); /* disable PORTB interrupts */
    if(inb(INTA01) & BIT0) outb(INTA01, par.int_mask); /* restore 8259 mask */
    fprintf(stderr, "%s 0, msg);
    exit(-1);                /* abnormal program termination */
}
```

```

;*****
title    EXTERNAL INTERRUPT HANDLER - ext_int
;*****
;*****

include param.asm

        codeseg segment para    public  'code'
        assume cs:codeseg
        public ext_int_,ext0_p_,ext1_p_,ext_f_

ext0_p_    label    byte                ;access offset & segment using C pointer
ext0_off dw    0                ;offset of ext0 buffer
ext0_seg   dw    0                ;segment of ext0 buffer

ext1_p_    label    byte                ;access offset & segment using C pointer
ext1_off dw    0                ;offset of ext1 buffer
ext1_seg   dw    0                ;segment of ext1 buffer

ext_f_     label    byte                ;ext int flag = # external int lines
ext_f      db    0

ext_int_ proc    far                ;entry point for interrupt (ext_int in C)

        pushreg                    ;save ax, di, ds, dx, es

        cmp     cs:ext_f,F_EXT0                ;Don't bother to read PORTC if only
        je      get_ex0                        ;1 external interrupt line

        mov     dx,PORTC
        in      al,dx                ;get port C input lines
        test    al,F_EXT0            ;check if PORTC bit F_EX0 is high
        jz      ck_ex1

get_ex0:
        gettime cs:ext0_off            ;store time of interrupt in ext0 buffer
        mov     cs:ext0_off,di        ;inc offset to point to next free byte
        cmp     di,BUF_EXT0          ;check if next write will overflow buf
        jb      ext_int_exit

mask_off:
        in      al,INTA01            ; get current interrupt mask
        or      al,PORTB_IRQ_BIT     ; mask off further interrupts
        out     INTA01,al
        jmp     short ext_int_exit

ck_ex1:
        test    al,F_EXT1            ;check if PORTC bit F_EX1 is high
        jz      ext_int_exit

```

```

        getime cs:ext1_off      ;store time of interrupt in ext1 buffer
        mov    cs:ext1_off,di   ;inc offset to point to next free byte
        cmp    di, BUF_EXT1     ;check if next write will overflow buf
        jnb    mask_off

ext_int_exit:
        mov     dx,PORTB
        in      al,dx            ;get current stimulus
        out     dx,al           ;reset PORTB interrupt
        mov     al,EOI
        out     INTA00, al      ;acknowledge interrupt to 8259

        popreg                  ;restore ax, di,ds,dx,es

        iret

ext_int_      endp

codeseg ends

        end

```

/******

FUNCTION SUBROUTINE GETLINE(s)

PURPOSE:

getline will get a line of input and return the number of
characters in the input line. 0 = blank line

INPUT:

s = character array of length MAXLINE

OUTPUT:

s = input line from standard input

RETURNS:

length of input line

0 = carriage return only - no blanks or characters

DEPENDENCIES:

param.h

stdio.h

*****/

```
#include <stdio.h>
```

```
#include "param.h"
```

```
int getline(s)
```

```
char s[];
```

```
{
```

```
    int c,
```

```
        i = 0,
```

```
        lim = MAXLINE;
```

```
    while(--lim > 0 && (c = getchar()) != EOF && c != '\0') s[i++] = c;
```

```
    s[i] = '\0';
```

```
    return(i);
```

```
}
```

```
/******
```

SUBROUTINE - GETPAR(fp)

```
*****
```

PURPOSE:

get the parameter struct from the default file

INPUT:

fp - filename containing parameters for input
external - par: parameter structure defined in param.h

OUTPUT:

external - par (modified)

DEPENDENCIES:

getline() - function used to read in 1 line at a time
stdio.h
math.h
param.h

```
*****/
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include "param.h"
```

```
void getpar(fp)
```

```
FILE *fp; /* parameter file pointer */
```

```
{
```

```
    extern PAR_STRUCT par; /* parameter structure */
```

```
    int c0,c1; /* temp byte to make stimulus word */
```

```
    int i; /* temporary index */
```

```
    if(fscanf(fp,"%d",&par.nchan) == 0) /* number of channels */  
        error("getpar: unable to read par.nchan");
```

```
    if(fscanf(fp,"%d",&par.response) == 0) /* # responses expected */  
        error("getpar: unable to read par.response");
```

```
    if(fscanf(fp,"%d",&par.external) == 0) /* # ext. interrupts PORTB_IRQ */  
        error("getpar: unable to read par.external");
```

```
    if(fscanf(fp,"%d",&par.ntrial) == 0) /* number of trials */  
        error("getpar: unable to read par.ntrial");
```

```
    if(fscanf(fp,"%d",&par.s_per) == 0) /* sample period in usec */  
        error("getpar: unable to read par.s_per");
```

```

if(fscanf(fp,"%d",&par.wt_dur) == 0) /* warning tone duration msec */
    error("getpar: unable to read par.wt_dur");

if(fscanf(fp,"%d",&par.s_total) == 0) /* number of sample pts */
    error("getpar: unable to read par.s_total");

if(fscanf(fp,"%d",&par.s_stim0) == 0) /* stim 0 on after stim # */
    error("getpar: unable to read par.s_stim0");

/* GET TRIAL DEPENDENT DATA & CHECK RANGE ON DELAYS */

for (i=0; i < par.ntrial; i++){
    if (fscanf(fp,"%x %x %d %d %d %d",
        &c0, &c1, /* stimulus is hex pair (two each) */
        /* 1st two = stim0, 2nd two = stim1 */
        &par.s_isi[i], /* # sampls after stim0 until stim1 */
        &par.delays[i], /* delay is in msec */
        &par.wt_pers[i], /* wt_pers = period in usec */
        &par.wt_fore[i]) < 6) /* wt_fore = foreperiod in msec */
        error("getpar: unable to read trial dependent data");
    par.stims[i] = c0 | ((c1 << 8) & ~0xff);
}

fclose(fp); /* CLOSE FILE */
return;
}

```



```
/******
```

PROGRAM - MAKEPAR

```
*****
```

PURPOSE:

make a parameter file to be used by the dac program
user is prompted for output file name & various parameters

DEPENDENCIES:

getline() - used to read in one line at a time
writepar() - used to write out parameter file
error() - standard error writing routine
stdio.h
param.h

```
*****/
```

```
#include <stdio.h>
```

```
#include "param.h"
```

```
#define getp(form,v) if(getline(s) != 0)sscanf(s,form,v)
```

```
PAR_STRUCT par = {5,2,2,0,0,0,101, /* parameter structure */  
0,4167,500,600,120};
```

```
char pc = 'n'; /* pc = y practice case, else pc = n */  
char s[MAXLINE]; /* string for inputting parameters */  
int i, seed; /* temporary index & seed for random */  
int rnd,temp; /* temp variables for randomizing */  
int nchoice = 4; /* number of choices in this block */  
int isi = 15; /* ISI in # of sample pts */  
int wtper2 = 100; /* 2 choice case has low warn tone */  
int wtper4 = 1000; /* 4 choice case has high warn tone */  
int delay[MAX_TRIAL]; /* temp for randomizing delay */  
int wtfore[MAX_TRIAL]; /* temp for randomizing fore periods */  
int stim_wt[MAX_TRIAL][2]; /* temp for randomizing stimuli & wt */  
int smask2[2] = 0x204,0x402; /* stimulus mask for 2 choice case */  
int smask4[12] = 0x102,0x104,0x108, /* stimulus mask for 4 choice case */  
0x201,0x204,0x208,  
0x401,0x402,0x408,  
0x801,0x802,0x804;  
int min_delay = 3500;  
int max_delay = 3500; /* min & max ISI delays */  
int wmask[3] = 1500,2000,2500; /* choice of these wt foreperiod */  
FILE *fp; /* file pointer for output file */
```

```
main()  
{
```

```

printf("Enter the output parameter file name ");
getp("%s",s);
if((fp = fopen(s,"w") ) == NULL){
    printf("unable to open output file name: %s 0,s);
    exit(-1);
}

s[0] = 'y';
while(!s[0] == 'N' || s[0] == 'n'){

    printf("Enter new value or CR to keep old 0);

    printf("number of channels = %d ?",par.nchan);
    getp("%d",&par.nchan);

    printf("number of external interrupt lines (0,1or 2) = %d ?",par.external);
    getp("%d",&par.external);

    printf("number of trials (max = %d) = %d ?",MAX_TRIAL,par.ntrial);
    getp("%d",&par.ntrial);

    printf("sampling period in usec (max = 64000) = %d ?",par.s_per);
    getp("%d",&par.s_per);

    printf("warning tone duration in msec = %d ?",par.wt_dur);
    getp("%d",&par.wt_dur);

    printf("number of samples per trial = %d ?",par.s_total);
    getp("%d",&par.s_total);

    printf("sample # after which to turn stimulus 0 on = %d ?",par.s_stim0);
    getp("%d",&par.s_stim0);

    printf("Trial min,max delay (msec) = %d, %d ?",min_delay,max_delay);
    if(getline(s) != 0)
        sscanf(s,"%d %d",&min_delay, &max_delay);

    printf("Foreperiod values (msec) = %d, %d, %d ?",wmask[0],wmask[1],wmask[2]);
    if(getline(s) != 0)
        sscanf(s,"%d,%d,%d",&wmask[0],&wmask[1],&wmask[2]);

    printf("Single stimulation practice block (y or n) = %c ?",pc);
    getp("%c",&pc);

    if(pc == 'y' || pc == 'Y'){
        par.response = 1;          /* one light so one response */
        isi = 32765;              /* second stim will not come on */
    } else {
        printf("ISI in # of sample pts (1 pt = %d usec) = %d ?",par.s_per,isi);
        getp("%d",&isi);
    }
}

```

```

printf("Two choice (2), mixed (3) or four choice (4) exp. = %d ?",nchoice);
getp("%d",&nchoice);

printf(" Oview Parameters Again ?");
getline(s);                                /* return to top of loop */
}

printf("Enter a seed for the random number generator ?");
scanf("%d",seed);
srand(seed);                               /* initialize random number generator */

for(i=0; i< par.ntrial; i++){
    delay[i] = min_delay + (long) rand() * (max_delay - min_delay) / 32767;
    wtfore[i] = wmask[i % 3];
}

for(i=0; i< par.ntrial; i++){               /* randomize wt_fore periods */
    rnd = (long) rand() * par.ntrial / 32767;
    if( rnd == par.ntrial) rnd--;
    temp = wtfore[i];
    wtfore[i] = wtfore[rnd];
    wtfore[rnd] = temp;
}

switch (nchoice) {

    case 2:                                /* two trial case */
        for(i=0; i<par.ntrial; i++){
            stim_wt[i][0] = smask2[ i % 2];
            stim_wt[i][1] = wtper2;
        }
        break;

    case 3:
        for(i=0; i < par.ntrial; i += 2){
            stim_wt[i][0] = smask2[ (i/2) % 2];
            stim_wt[i][1] = wtper2;
            stim_wt[i+1][0] = smask4[ (i/2) % 12];
            stim_wt[i+1][1] = wtper4;
        }
        break;

    case 4:
        for(i=0; i < par.ntrial; i++){
            stim_wt[i][0] = smask4[ i % 12];
            stim_wt[i][1] = wtper4;
        }
        break;

    default:
        printf("Error, nchoice != 2,3 or 4 0);
        exit(-1);

```

```

}

for(i=0; i< par.ntrial; i++){
    /* randomize stimuli */
    rnd = (long) rand() * par.ntrial / 32767;
    if( rnd == par.ntrial) rnd--;
    temp = stim_wt[i][0];
    stim_wt[i][0] = stim_wt[rnd][0];
    stim_wt[rnd][0] = temp;
    temp = stim_wt[i][1];
    stim_wt[i][1] = stim_wt[rnd][1];
    stim_wt[rnd][1] = temp;
}

rnd = (pc == 'y' || pc == 'Y')? 0x00ff : 0xffff;

for(i= 0; i<par.ntrial; i++){
    /* copy to parameter structure */
    par.stims[i] = stim_wt[i][0] & rnd;
    par.s_isi[i] = isi;
    par.delays[i] = delay[i];
    par.wt_pers[i] = stim_wt[i][1];
    par.wt_fore[i] = wtfore[i];
}

/* CHECK RANGE OF PARAMETERS */

if(par.response > 2) error("makepar: par.response > 2");
if(par.external > 2) error("makepar: par.external > 2");
if(par.ntrial > MAX_TRIAL) error("makepar: par.ntrial > MAX_TRIAL ");
for(i=0; i<par.ntrial; i++){
    if(par.delays[i] < par.wt_dur + par.wt_fore[i])
        error("makepar: par.delays < par.wt_dur + par.wt_foreperiod");
    if(par.s_isi[i] == 0)
        error("makepar: par.s_isi == 0");
    if(par.s_isi[i] + par.s_stim0 == par.s_total)
        error("makepar: par.s_isi + par.s_stim0 == par.s_total");
}

writepar(fp);
fclose(fp);

return;
}

```

```
.....
INCLUDE FILE - PARAM.ASM
.....
```

```
CONTAINS ALL NECESSARY DEFINITIONS FOR TECMAR BOARD
```

```
IMPLEMENTED AS AN I/O DEVICE AT 0710H - 071FH
```

```
R/W AT THE END OF THE COMMENTS REFER TO READ & WRITE
.....
```

```
TECMAR EQU 0710H; /* TECMAR STARTING ADDRESS */
```

```
; /* DIGITAL / ANALOG CONVERTERS */
```

```
LB_0DA EQU 0710H; /* LBYTE OF D/A # 0 W */
```

```
HB_0DA EQU 0711H; /* HBYTE OF D/A # 0 W */
```

```
LB_1DA EQU 0712H; /* LBYTE OF D/A # 1 W */
```

```
HB_1DA EQU 0713H; /* HBYTE OF D/A # 1 W */
```

```
; /* GENERAL CONTROL */
```

```
CONTROL EQU 0714H; /* CONTROL BYTE W */
```

```
STATUS EQU 0714H; /* STATUS BYTE R */
```

```
; /* ANALOG / DIGITAL CONVERTERS */
```

```
CHAN_AD EQU 0715H; /* A/D INPUT CHANNEL # W */
```

```
CONV_AD EQU 0716H; /* START A/D CONVERSION W */
```

```
LB_AD EQU 0715H; /* LOW BYTE OF A/D R */
```

```
HB_AD EQU 0716H; /* HIGH BYTE OF A/D R */
```

```
; /* TIMER CONTROLS */
```

```
T_ACK EQU 0717H; /* TIMER INT ACKNOWLEDGE W */
```

```
T_DATA EQU 0718H; /* TIMER DATA R/W */
```

```
T_CONT EQU 0719H; /* TIMER CONTROL BYTE R/W */
```

```
; /* PARALLEL PORT CONTROLS */
```

```
PORTA EQU 071CH; /* PORT A R/W */
```

```
PORTB EQU 071DH; /* PORT B R/W */
```

```
PORTC EQU 071EH; /* PORT C R/W */
```

```
P_CONT EQU 071FH; /* PORT CONTROL BYTE W */
```

```
; /* USEFUL TECMAR DEFINITIONS THIS APPLICATION
```

```

SAVES34      EQU 0ach      ;TCONT: save timer 3&4 counts in hold regs 1&2
H_CT3 EQU 1bh      ;TCONT: T DATA -> hold reg 3 & cycle hold reg
DISAS5 EQU 0d0h      ;TCONT: disable timer 5
CLRT5 EQU 0e5h      ;TCONT: clear - reset timer 5 high (TC low)
STEPT5      EQU 0f5h      ;TCONT: increment timer 5

```

```

;.....

```

``` ; GENERAL DEFINITIONS ```

```

;.....

```

```

PORTB_IRQ_BIT EQU 08h      ;PORTB IRQ3 -- 4th bit in OCW1 of 8259, check - param.c

```

```

;.....
;MAXIMUM EXT BUF SIZE IS 0FFFC & MUST BE MULT OF 4 */
BUF_EXT0 EQU 0ffch      ;MAX NUMBER OF EXTERNAL 0 INTERRUPTS == BUF_EXT0 / 4
BUF_EXT1 EQU 0ffch      ;max number of external 1 interrupts == BUF_EXT1 / 4
F_EXT0      EQU 40h      ;PORTC bit C6 is used for external 0 input
F_EXT1      EQU 80h      ;PORTC bit C7 is used for external 1 input

```

```

;.....

```

``` ; INTERRUPT CONTROLLER DEFINITIONS ```

```

;.....

```

```

EOI EQU 20h      ;end of interrupt command for 8259
INTA00EQU 20h      ;1st 8259 interrupt controller port 1
INTA01EQU 21h      ;1st 8259 interrupt controller port 2
INTB00EQU 0a0h      ;2nd 8259 interrupt controller port 1
INTB01EQU 0a1h      ;2nd 8259 interrupt controller port 2
INT_TYPE EQU 70h      ;start of interrupt vectors for 2nd 8259 (IRQ 8-15 )

```

```

;.....

```

``` ; STRUCTURE DEFINITIONS ```

```

;.....

```

```

; IC_STRUC flag bit definition

```

```

F_RESP0      EQU 01h      ;waiting for response 0
F_RESP1      EQU 02h      ;waiting for response 1
F_AD EQU 10h      ;perform A/D conversion
F_DELAY      EQU 40h      ;delay finished
F_SAMP      EQU 80h      ;sampling finished

```

```

IC_STRUC      struc      ; this definition corresponds to C struct of same name

```

```

nchan db 0      ;number of channels to be converted
flags db 0      ;control byte for interrupt routine
stim0 db 0      ;stimulus zero to be sent to parallel port out

```

```

stim1 db 0 ;stimulus one to be sent to parallel port out
s_stim0 dw 0 ;sample number to turn stimulus 0 on
s_stim1 dw 0 ;sample number to turn stimulus 1 on
s_total dw 0 ;total number of samples
s_cur dw 0 ;current sample number (set to 0 initially)
buf_off dw 0 ;buffer offset value for next data location
buf_seg dw 0 ;buffer segment location
res_off dw 0 ;response structure offset for next data location
res_seg dw 0 ;response structure segment location

```

```
IC_STRUC ends
```

```

; BYTE OFFSETS TO MATCH ----- RESPONSE STRUCTURE IN C
;typedef struct {
STIME0 EQU 0 ; unsigned long int stime0,
RTIME0 EQU 4 ; rtime0,
RTIME1 EQU 8 ; rtime1;
RESP0 EQU 12 ; unsigned char resp0,
RESP1 EQU 13 ; resp1; } RESP_STRUCT

```

```
*****
```

```
; MACRO DEFINITIONS
```

```
*****
```

```

pushreg macro ;push register macro
    push ax
    push di
    push ds
    push dx
    push es
endm

```

```

popreg macro ;pop register macro
    pop es
    pop dx
    pop ds
    pop di
    pop ax
endm

```

```

gettime macro x_off ;get time from timers 3 & 4 al,dx,di,es used
    mov al, SAVES34
    mov dx, T_CONT
    out dx,al ;save counts in hold reg
    mov al, H_CT3
    out dx,al ;T_DATA -> timer1 hold reg & cycle hold reg

```

les	di, dword ptr x.off	;esi:di <- x.off:x_seg
mov	dx, T_DATA	;dx <- T_DATA reg
insb		;esi:di <- timer 3 hold reg, inc di
insb		
insb		;esi:di <- timer 4 hold reg, inc di
insb		
endm		

AD-A188 205

ELECTROMAGNETIC METRICS OF MENTAL WORKLOAD(U) PURDUE
UNIV LAFAYETTE IN EEG SIGNAL PROCESSING LAB
J I AUNON ET AL SEP 87 AFOSR-TR-87-1663 AFOSR-85-0313

3/3

UNCLASSIFIED

F/G 5/8

NL





MICROCOPY RESOLUTION TEST CHART

```
/******
```

INCLUDE FILE - PARAM.H

```
*****
```

includes all parameter and macro definitions for DAC software

```
*****/
```

```
/******
```

MACRO DEFINITIONS

```
*****/
```

```
#define out2b(p,d) outb(p,d); outb(p,(d)>>8); /* out 2 bytes , low first */
#define min(a,b) ( (a) > (b) ? (b) : (a) ) /* minimum function */
#define max(a,b) ( (a) > (b) ? (a) : (b) ) /* maximum function */
```

```
/******
```

GENERAL CONSTANT DEFINITIONS

```
*****/
```

```
#define MAX_TRIAL 200                    /* max number trials - can change */
#define MAXLINE 80                    /* max number of chars/line - can change */
#define HIGH_FREQ 3000                /* threshold for high freq line to go high */
```

```
#define BIT0 0x01                    /* bit selectors */
#define BIT1 0x02
#define BIT2 0x04
#define BIT3 0x08
#define BIT4 0x10
#define BIT5 0x20
#define BIT6 0x40
#define BIT7 0x80
```

```
/******
```

INTERRUPT STRUCT FLAG DEFINITIONS

```
*****/
```

/* IC_STRUCT ic.flags bits */

```
#define F_RESP0 BIT0                /* waiting for response 0 */
#define F_RESP1 BIT1                /* waiting for response 1 */
#define F_AD BIT4                    /* perform a/d conversion */
```

```
#define F_DELAY      BIT6          /* delay finished */
#define F_SAMP       BIT7          /* sampling this trial finished */
/*****
```

STRUCTURE DEFINITIONS

```
*****/
```

```
typedef struct {          /* INT_STRUCT (interrupt) structure */

    int      intnum;        /* interrupt request line IRQ # */
    unsigned i_oldseg;      /* segment from old interrupt vector */
    unsigned i_oldoff;      /* offset from old interrupt vector */
    void      (*i_new)();    /* name of new routine for interrupt */
}INT_STRUCT;
```

```
typedef struct {          /* RESP_STRUCT response structure type */
                                /* MUST MATCH param.asm byte offsets */

    unsigned long int stime0,    /* time of stimulus 1 being turned on */
                rtime0,        /* time of response to stimulus 0 */
                rtime1;        /* time of response to stimulus 1 */
                                /* response time 0 = rtime0 - stime0 */
    unsigned char resp0,        /* response byte for stimulus 0 */
                resp1;         /* response byte for stimulus 1 */
} RESP_STRUCT;
```

```
typedef struct {          /* IC (interrupt control) structure type */
                                /* must match IC assembler structure EXACTLY */

    unsigned char          /* same as define byte in asm */
        nchan,            /* number of channels to convert */
        flags;            /* control byte on interrupt entry */
                                /* see above interrupt flag definitions */
    unsigned int           /* same as define word in asm */
        stim,             /* lbyte = stimulus 0, hbyte = stimulus 1 */
        s_stim0, /* after samp# = s_stim0 - send stimulus 0 */
        s_stim1, /* after samp# = s_stim1 - send stimulus 1 */
                                /* stimulus 1 = current stimulus, stim 1 */
                                /* correct resp0 will clear stim 0 */
        s_total, /* total number of sample points per trial */
                                /* 1st samp=samp# 0, stim cleared after s_total */
                                /* s_stim0, s_stim1, s_total must be all diff */
        s_cur;            /* current sample number (initially 0) */
    unsigned char
        *buf;             /* (low word) offset of data buffer */
                                /* (high word) segment of data buffer */
}RESP_STRUCT
```

```

        *resp;          /* (low word) offset of resp struct */
                        /* (high word) segment of resp struct */

} IC_STRUCT;

typedef struct {          /* PAR_STRUCT (parameter) structure type */

    unsigned char
        nchan,            /* total number of channels */
        response,         /* # of responses expected, 0,1 or 2 */
        external,         /* flag if ext int on PORTC using PORTB_IRQ */
        reject,           /* flag to reject previous trial */
        int_mask,         /* old 8259 interrupt mask */
        *bufs[2];         /* data buffers for temporary storage */

    unsigned int
        ntrial,           /* number of trial (stimuli) to be presented */
        itrial,           /* current trial */
        s_per,            /* sampling period in usec (F1) */
        wt_dur,           /* warning tone duration in msec (F4) */
        s_total,          /* number of samples to be taken */
        s_stim0,           /* sample number to turn stimulus 0 on */
        stims[MAX_TRIAL], /* stimulus - lbyte = stim0, hbyte=stim1 */
        s_isi[MAX_TRIAL], /* # of samples after stim0 to turn stim1 on */
        delays[MAX_TRIAL], /* trial delay between sampling in msec (F4) */
        wt_fore[MAX_TRIAL], /* wt fore period = ms after tone until samp */
        wt_pers[MAX_TRIAL]; /* warning tone period in usec (1/f) F1 */

    int
        fd_dat,           /* file descriptor for data output */
        fd_ext0,          /* file descriptor for ext 0 interrupt times */
        fd_ext1;          /* file descriptor for ext 1 interrupt times */

    FILE
        *fp_res,          /* file pointer for response file */
        *fp_doc;          /* file pointer for documentation file */

} PAR_STRUCT ;

```

```

;*****
title  RESPONSE INTERRUPT HANDLER - res_int
;*****
;*****

include param.asm
extrn  ic:far

codeseg segment para  public  'code'
assume cs:codeseg, ds:seg ic
public res_int_

res_int_ proc  far                ;entry point for interrupt (res_int in C)

    pushreg                      ;save ax, di, ds, dx, es
    mov  al, EOI                 ;acknowledge interrupt to 8259
    out  INTA00, al              ;
    mov  ax, seg ic              ;load dataseg
    mov  ds, ax
    les  di, dword ptr ic.res_off ;es:di -> resp structure

    mov  dx, T_CONT              ; dx -> timer control port
    mov  al, SAVES34             ;save count in timer 3 & 4 in hold reg
    out  dx, al
    mov  al, H_CT3               ;T_DATA reg -> timer 3 hold reg & cycle
    out  dx, al
    mov  dx, PORTA               ;dx -> input port
    in   al, dx                  ;al = response - reset 8255 IBF latch

    test ic.flags, F_RESP0       ;determine if waiting for response 0
    js   check1

    xor  ic.flags, F_RESP0       ;clear waiting for resp0 flag
    mov  es:[di][RESP0], al      ;store response 0
    add  di, RTIME0              ;es:di -> response time 0
    cmp  al, ic.stim0            ;check if right response
    je   clear_stim              ;equal so clear stim & save time
    jmp  short save_time         ;save response time only

check1:
    test ic.flags, F_RESP1       ;waiting for response 1
    js   res_int_exit            ;not waiting for response

    cmp  al, es:[di][RESP0]      ;not valid if same as response 0
    je   res_int_exit            ;

    xor  ic.flags, F_RESP1       ;clear waiting for response 1 flag
    mov  es:[di][RESP1], al      ;save response
    add  di, RTIME1              ;es:di -> response time 1

```

```

        cmp     al, ic.stim1      ;check if response 1 = stimulus 1
        jne     save_time        ;save time only

clear_stim:
        mov     ah,al            ;ah = current response
        not     ah               ;make clear mask
        mov     dx, PORTB        ;dx -> output port
        in      al,dx            ;al = current stimulus
        and     al,ah            ;clear response from current stimulus
        out     dx,al            ;CAUTION: This will reset PORTB OBF & INTR

save_time:
        mov     dx, T_DATA       ;T_DATA reg -> timer 2 hold reg
        insb                     ;store current time from timer 1
        insb
        insb                     ;store current time from timer 2
        insb

res_int_exit:

        popreg                  ;restore ax, di,ds,dx,es

        iret

res_int_endp

codeseg ends

end

```

/*.....

SUBROUTINE - RESET_INT(ip)

.....

PURPOSE:

reset an interrupt vector to the value stored in the interrupt struct

INPUT:

ip - INTERRUPT STRUCTURE POINTER

OUTPUT:

none

DEPENDENCIES:

param.h -- general parameters

stdio.h -- C standard input/output

dos.h ---- C interface to dos routines

...../

#include <stdio.h>

#include <dos.h>

#include "param.h"

void reset_int(ip)

INT_STRUCT *ip;

/* interrupt info structure */

{

struct reg r;

/* register structure for DOS int */

r.r_ds = ip->i_oldseg;

r.r_dx = ip->i_oldoff;

r.r_ax = SETINT | ip->intnum;

/* offset & pointer */

intcall (&r, &r, DOSINT);

/* redirect interrupt */

return;

}

/******

FUNCTION SUBROUTINE SAMP()

PURPOSE:

Perform all necessary operations to start the tecmar board sampling data for each new trial.

- 1) check if last trial to be rejected
- 1) check if all trials are finished
- 2) delay finished flag set -> call error & terminate
inter trial delay finished before high level processing = fatal error
- 3) increment all ic parameter values
- 4) wait for delay finished flag to be set
- 5) reset delay finished flag
- 6) start sampling in background
- 7) copy data from previous trial to hard disk

INPUT:

none - external ic, par

OUTPUT:

data from previous buffer to output file

RETURN (int):

- 1 = normal delay - sampling initiated this trial - stored previous
- 0 = normal finish - no sampling initiated - last two trials stored

DEPENDENCIES:

tm_int() - performs actual sampling in background
error() - error exit & terminate routine
ic - interrupt delay structure (external)
par - experiment parameter structure (external)
resps - response structure for storing responses & times (external)
tecmar.h - parameters for tecmar board
param.h - general parameters
stdio.h - C standard input/output

*****/

```
#include <stdio.h>
#include "tecmar.h"
#include "param.h"
```

```
int samp()
```

```
{
    extern PAR_STRUCT par;          /* experiment parameter structure */
    extern IC_STRUCT ic;           /* interrupt delay structure */
    extern RESP_STRUCT resps[];    /* storage for responses & times */
}
```

```

/* CHECK REJECT TRIAL FLAG & IF FINISHED - RETURN (0) */

    if (par.reject)
        par.itrial--; /* reject - decrement trials */

    else if (par.itrial == par.ntrial){ /* check if done */
        write(par.fd_dat, par.bufs[(par.itrial-1)&BIT0], 2*par.nchan*par.s_total);
        close(par.fd_dat); /* close output data file */
        return(0); /* all finished return 0 */
    }

/* CHECK IF DELAY HAS FINISHED YET */

    if(ic.flags & F_DELAY) error("startsamp: delay finished early ");

/* INCREMENT PARAMETERS IN INTERRUPT CONTROL STRUCTURE ic */

    ic.buf = par.bufs[par.itrial & BIT0]; /* next buffer */
    ic.s_stim1 = par.s_stim0 + par.s_isi[par.itrial]; /* samp # for stim1 */
    ic.stim = par.stims[par.itrial]; /* stimulus word */
    ic.s_cur = 0; /* current sample # */
    ic.resp = &resps[par.itrial]; /* storage for responses */

/* WAIT FOR DELAY TO END & THEN RESET FLAGS */

    while(!(ic.flags & F_DELAY)); /* wait */
    ic.flags = F_RESP0 | F_RESP1 | F_AD | (ic.flags & ~(F_DELAY));
    /* clear delay flag, set a/d , waiting for response flags */

/* Produce sync pulse on port b (5th bit) for ext sync of scope */
    outb(PORTB, 0x10);

/* START SAMPLING */
    outb(T_CONT, T5); /* T_DATA -> timer 5 mode reg */
    out2b(T_DATA, TM5us); /* set for usec delay */
    out2b(T_DATA, par.s_per); /* set count down sampling freq divider */
    outb(T_CONT, A_LSX | S5); /* start sampling - arm & load timer 5 */

/* STORE DATA FROM PREVIOUS TRIAL IF NO REJECT FLAG */

    if (par.reject){
        par.reject = 0; /* reset reject flag */
    }else{ /* store previous trial */
        write(par.fd_dat, par.bufs[(par.itrial-1)& BIT0], 2*par.nchan*par.s_total);
    }

/* RETURN NORMALLY */

    return (1);
}

```

/******

SUBROUTINE - SET_INT(ip)

PURPOSE:

redirect an interrupt vector to an interrupt routine

INPUT:

ip - INTERRUPT STRUCTURE POINTER

OUTPUT:

none

DEPENDENCIES:

param.h -- general parameters
stdio.h -- C standard input/output
dos.h ---- C interface to dos routines

*****/

#include <stdio.h>

#include <dos.h>

#include "param.h"

void set_int(ip)

INT_STRUCT *ip; /* interrupt info structure */

```
{
    struct reg r; /* register structure for DOS int */
    r.r_ax = GETVEC | ip->intnum;
    intcall (&r, &r, DOSINT); /* get old interrupt vector */
    ip->i_oldseg = r.r_es; /* save old int seg & offset */
    ip->i_oldoff = r.r_bx;
    ptoreg(csreg, r.r_dx, r.r_ds, ip->i_new); /* convert procedure name to */
    r.r_ax = SETINT | ip->intnum; /* offset & pointer */
    intcall (&r, &r, DOSINT); /* redirect interrupt */
    return;
}
```

/******

SUBROUTINE FUNCTION - SETUP()

PURPOSE:

prompt user for output file name & parameters
load parameter structure & open output files

INPUT:

prompt - fpar: filename containing default parameters for input
- fout: output file name (dir + fname (max 7 char))
several files will be produced with various extensions
external - par: parameter structure

OUTPUT:

fout0.dat - output binary data file
fout0.doc - output documentation file
fout0.par - output parameter file
fout0.res - output response file
fout0.ex0 - output external 0 interrupt times
fout0.ex1 - output external 1 interrupt times
external - par (modified)

DEPENDENCIES:

getpar.c - gets the parameters from a parameter file
checkpar.c - prompts the user for the parameter changes
writepar.c - writes the parameter file to disk
stdio.h
tecmar.h
param.h

*****/

```
#include <stdio.h>
#include "tecmar.h"
#include "param.h"
```

```
void setup()
```

```
{
    extern PAR_STRUCT    par;    /* parameter structure defined in param.h */

    char run,            /* run number */
        string[MAXLINE], /* temp storage for a string */
        fpar[MAXLINE],   /* parameter file name */
        fout[MAXLINE-7]; /* parameters must be in proper format */
                        /* output file name (max 7 char) */
                        /* various extensions will be appended */
}
```

```

FILE *fp;                /* general file pointer */

#define ECAL ".cal"        /* fout.cal - calibration file */
#define EDAT ".dat"        /* fout.dat - binary data file */
#define EDOC ".doc"        /* fout.doc - documentation file */
#define EPAR ".par"        /* fout.par - parameter file */
#define ERES ".res"        /* fout.res - response file */
#define EEX0 ".ex0"        /* fout.ex0 - external 0 int file */
#define EEX1 ".ex1"        /* fout.ex1 - external 1 int file */

/* PROMPT FOR FILE NAME */

printf("Enter file name in the following form drive:directory\fname 0);
printf("drive and directory are optional 0);
printf("fname must be <= 7 char (with NO .ext):");
while (getline(fout) == 0);

/* CHECK FOR EXISTENCE OF A CALIBRATION FILE */

sprintf(string,"%s%s",fout,ECAL); /* make cal file name (NO RUN #) */
if((fp = fopen(string, "r")) == NULL)
    fprintf(stderr,"setup: WARNING no cal file with matching name0);
else
    fclose(fp);

/* DETERMINE RUN NUMBER - SET STRING -> OUTPUT PARAMETER FILE */

for(run = '0'; run <= '9'; run = run + 1){ /* max of 10 runs */
    sprintf(string,"%s%c%s",fout,run,EPAR);
    if((fp = fopen(string, "r")) == NULL) break;
    fclose(fp);
}
if (run > '9') error("setup: more than 9 runs with this name");

/* ENTER DEFAULT PARAMETER FILE OR PROMPT FOR VALUES */

printf("Enter file name for default parameters (or return):");

if(getline(fpar) == 0) /* allow user to review & change default par */
    checkpar();
else{ /* get values from default file */
    if((fp = fopen(fpar, "r")) == NULL)
        error("setup: unable to open input parameter file");
    getpar(fp);
    checkpar(); /* allow user to review parameters */
    fclose(fp);
}

/* WRITE OUT PARAMETER FILE */

if ((fp = fopen(string, "w")) == NULL)

```

```

        error("setup: unable to open output parameter file");

writepar(fp);          /* write out parameter file & close */
fclose(fp);

/* OPEN OUTPUT DATA FILE (BINARY) */

sprintf(string,"%s%c%s",fout,run,EDAT); /* make data filename */
if ((par.fd_dat = creat(string,1)) == -1)
    error("setup: unable to create data file");

/* OPEN RESPONSE FILE IF NECESSARY */

if(par.response){
    sprintf(string,"%s%c%s",fout,run,ERES); /* make response filename */
    if((par.fp_res = fopen(string,"w")) == NULL)
        error("setup: unable to create response file");
}

/* OPEN EXTERNAL FILES IF NECESSARY */

if(par.external > 0){
    sprintf(string,"%s%c%s",fout,run,EEX0); /* make external 0 filename */
    if((par.fd_ext0 = creat(string,1)) == -1)
        error("setup: unable to create external 0 file");
}

if(par.external == 2){
    sprintf(string,"%s%c%s",fout,run,EEX1); /* make external 1 filename */
    if((par.fd_ext1 = creat(string,1)) == -1)
        error("setup: unable to create external 1 file");
}

/* OPEN DOCUMENTATION FILE & ALLOW USER TO INPUT DOCUMENTATION */

sprintf(string,"%s%c%s",fout,run,EDOC); /* make documentation fname */
if((par.fp_doc = fopen(string,"w")) == NULL)
    error("setup: unable to create documentation file");
printf("Enter documentation for this run here (CR to end): 0);
printf("Record A/D channels & externals here also 0);
while(getline(string) != 0)fprintf(par.fp_doc,"%s0,string);
printf("documentation written to %s%c%s 0, fout,run,EDOC);

/* RETURN */

return;

}

```

```

;*****
;
;      STORAGE.ASM
;*****

include param.asm

buffer0 segment para . public 'data'
    public buf0_
    buf0_ db      0fffh    dup (?)
buffer0 ends

buffer1 segment para    public 'data'
    public buf1_
    buf1_ db      0fffh    dup (?)
buffer1 ends

external0 segment para public 'data'
    public ext0, ext0_
    ext0_ label byte
    ext0  db      BUF_EXT0    dup (0)
external0 ends

external1 segment para public 'data'
    public ext1, ext1_
    ext1_ label byte
    ext1  db      BUF_EXT1    dup (0)
external1 ends

dataseg segment para    public 'data'
    public ic, ic_
    ic_   label byte          ;access this structure as ic in C
    ic    IC_STRUC <>        ;allocate space for id struc
dataseg ends

    end

```

/*****

SUBROUTINE - STORE()

*****/

PURPOSE:

Write responses, external interrupt times, documentation files
Close all files.

INPUT:

external par -- contains all file pointers
external ext0,ext1 -- buffers containing ext int times
external resps -- array of response structures

OUTPUT FILES:

fout?.doc -- documentation file
fout?.res -- response file if selected
fout?.ex0 -- ext int line 0 times if selected
fout?.ex1 -- ext int line 1 times if selected

DEPENDENCIES:

getline() - subroutine to get one line at a time
stdio.h
param.h

*****/

```
#include <stdio.h>
#include "param.h"
#include "tecmar.h"
```

```
void store()
```

```
{
    extern PAR_STRUCT    par;          /* external parameter structure */
    extern RESP_STRUCT    resps[]; /* external response structure */
    extern void ext0(),ext1(); /* external int buffers (NOT func) */
    extern unsigned long int
        *ext0_p,*ext1_p; /* pointers to end of ext int buffer */
    long int    r0,r1; /* computed response times */
    int    i; /* temporary index */
    char    string[80]; /* temporary string */
```

```
/* DOCUMENTATION FILE */
```

```
printf("Enter any additional documentation here (CR to end):0);
while(getline(string) != 0)printf(par.fp_doc,"%s0,string);
fclose(par.fp_doc);
```

```
/* RESPONSE FILE (five items per line) */
```

```
/* resp0. resp1, stim0 -> resp0 time, stim1 -> resp1 time, stim0 time */
```



```

if(par.response){
    for(i=0; i< par.ntrial; i++){
        /* set resp time to 0 if no response */
        r0 = (resps[i].rtime0 == 0)? 0 :resps[i].rtime0 - resps[i].stime0;
        r1 = (resps[i].rtime1 == 0)? 0 :resps[i].rtime1 - resps[i].stime0
            - (long int) par.s_isi[i] * par.s_per/100;
        fprintf(par.fp_res,"%x %x %ld %ld %lu0,
            resps[i].resp0,resps[i].resp1, r0, r1, resps[i].stime0);
    }
    fclose(par.fp_res);
}

/* EXTERNAL INTERRUPT LINE TIMES FILES (binary - long int) */

if(par.external){
    write(par.fd_ext0, ext0,
        sizeof(unsigned long int) * (ext0_p - (unsigned long int *)ext0));
    close(par.fd_ext0);
    if(par.external == 2){
        write(par.fd_ext1, ext1,
            sizeof(unsigned long int) * (ext1_p - (unsigned long int *)ext1));
        close(par.fd_ext1);
    }
}

return;
}

```

/******

SUBROUTINE - TEC_DONE()

PURPOSE:

deallocate tecmar board & reset interrupts

INPUT:

none - external par

OUTPUT:

none

DEPENDENCIES:

par ----- experiment parameter structure (external)
reset_int() -- reset the interrupt vector to its original value
error() -- error & terminate routine
tecmar.h - parameters for tecmar board
param.h -- general parameters
stdio.h -- C standard input/output

*****/

#include <stdio.h>

#include "param.h"

#include "tecmar.h"

void tec_done()

```
{
extern PAR_STRUCT par;          /* experiment parameter structure */
extern INT_STRUCT tm, res, ext; /* interrupt vector structures */

    outb(T_CONT, TIM_CLR);      /* reset all timers */
    outb(CONTROL, DISAUTO);     /* disable tecmar timer interrupts */
    reset_int(&tm);              /* reset timer interrupt vector */
    if(par.response){
        reset_int(&res);         /* reset response interrupt vector */
        outb(P_CONT, PC_RSET, INTE_AIN); /* reset PORTA interrupt enable */
    }
    if(par.external){
        reset_int(&ext); /* reset external interrupt vector */
        outb(P_CONT, PC_RSET, INTE_B); /* reset PORTB interrupt enable */
    }
    outb(INTA01, par.int_mask); /* reset interrupt mask */

    printf(" SAMPLING DONE - TECMAR & INTERRUPTS RESET 0);

    return;
}
```

/*****

SUBROUTINE - TEC_INIT()

*****/

PURPOSE:

initialize the tecmar board & start initial delay
Next step after this call should be the sampling loop

Tecmar Board Jumpers are assumed to be set at the following
timer 5 to generate interrupts (J7 pin 2 to pin 3)
timer interrupt to IRQ5 (J9 pin 14 to pin 5)
parallel port A interrupt to IRQ4 (J9 pin 12 to pin 4)
parallel port B interrupt to IRQ3 (J9 pin 13 to pin 3)

timer 1 is used to generate the warning tone gating for timer 2
timer 2 is used to generate the warning tone frequency
timers 3&4 are used to form a 32 bit real time clock
timer 5 is used to produce interrupts for sampling & delays

PORT A is used as the input port to read switches which cause interrupt
PORT B is used as the output port to turn lights or devices on
PORT C is used for handshaking on A&B and 2 input lines which
generate an "external" interrupt on Port B

INPUT:

external - ic, par, buf0, buf1

OUTPUT:

none

DEPENDENCIES:

set_int() - set interrupt vector by redirecting old vector
error() -- error & terminate routine
tecmar.h - parameters for tecmar board
param.h -- general parameters
stdio.h -- C standard input/output
dos.h -- C library for dos interface

*****/

```
#include <stdio.h>
#include <dos.h>
#include "param.h"
#include "tecmar.h"
#define INIT_DELAY 5000          /* program initial delay in msec */
```

```
void tec_init()
```

```
{
```

```

extern PAR_STRUCT par;          /* parameter structure */
extern IC_STRUCT ic;           /* interrupt control structure */
extern INT_STRUCT tm, res, ext; /* interrupt vector structures */
extern void buf0(), buf1();    /* buffers address (NOT functions) */
extern void ext0(), ext1();    /* ext int buffer address (NOT func) */
extern unsigned long int
    *ext0_p, *ext1_p;          /* pointers to ext int buffers */
extern char ext_f;             /* ext int flag = # interrupts */
extern void tm_int(),          /* timer interrupt handler */
    res_int(),                /* response interrupt handler */
    ext_int();                /* external interrupt handler */
extern unsigned char new_mask; /* new interrupt mask for 8259 */
extern int temp;

```

/** SET PARAMETERS IN PARAMETER & INTERRUPT CONTROL STRUCTURE **/

```

par.itrial = 0;                /* first trial is number 0 */
ic.nchan = par.nchan;          /* number of chanel */
ic.s_stim0 = par.s_stim0;      /* sample to turn stim 0 on */
ic.s_total = par.s_total;      /* number of samples */
par.bufs[0] = buf0;            /* set bufer pointers */
par.bufs[1] = buf1;

```

***** SET UP MASTER MODE REGISTER & TIMERS & WARNING TONE DURATION *****/

```

outb(T_CONT, TIM_CLR); /* master clear of all timers */

outb(T_CONT, MM_REG); /* T_DATA -> MM reg */
out2b(T_DATA, MM_SET); /* enable data pointer auto increment */

outb(T_CONT, T1);
out2b(T_DATA, TM1); /* set Timer 1 mode reg for warning gating */
outb(T_CONT, HOLD_TX | T1); /* T_DATA reg -> timer 1 hold reg */
out2b(T_DATA, par.wt_dur);

outb(T_CONT, T2); /* set Timer 2 mode reg for warning freq */
out2b(T_DATA, TM2);

outb(T_CONT, T3); /* set timers 3 & 4 to form 32 bit .1 msec */
out2b(T_DATA, TM3); /* counter for all program timing */
outb(T_CONT, T4);
out2b(T_DATA, TM4);

outb(T_CONT, T5); /* set Timer 5 mode reg for delay interrupts */
out2b(T_DATA, TM5ms); /* initially set for milli sec range */

```

/* CONFIGURE TECMAR BOARD */

```

outb(CONTROL, DISAUTO | TIME_INT); /* timer interrupt & no auto inc */
outb(P_CONT, PORT_SET); /* set up port control */

```

```

/* REDIRECT INTERRUPT VECTORS & MAKE NEW 8259 INTERRUPT MASK */

par.int_mask = inb(INTA01);          /* save old interrupt mask */
/* new mask = old + timer int - IRQ0 (sys clock) */
new_mask = ((~(1 << TM_IRQ) & par.int_mask) | BIT0);
tm.intnum = 8 + TM_IRQ;              /* Tech. Ref. page 5-5 */
tm.i_new = tm_int;                  /* tm.i_new -> timer int handler */
set_int(&tm);                       /* redirect int vector & store old */

if(par.response){                  /* responses expected */
    res.intnum = 8 + PORTA_IRQ;      /* Tech. Ref. page 5-5 */
    res.i_new = res_int;            /* res.i_new -> response int handler */
    set_int(&res);                  /* redirect int vector & store old */
    new_mask &= ~(1 << PORTA_IRQ); /* set mask to allow PORTA_IRQ */
    outb(P_CONT, PC_SET | INTE_AIN); /* enable input PORTA for interrupt */
}

if(par.external){                  /* ext interrupts using PORTB_IRQ */
    ext.intnum = 8 + PORTB_IRQ;      /* IBM/AT Tech. Ref. page 5-5 */
    ext.i_new = ext_int;            /* ext.i_new -> external int handler */
    set_int(&ext);                  /* redirect int vector & store old */
    new_mask &= ~(1 << PORTB_IRQ); /* set mask to allow PORTB_IRQ */
    outb(P_CONT, PC_SET | INTE_B); /* enable output PORTB for interrupt */
    ext0_p = ext0;                  /* set pointer to ext int buffer */
    ext1_p = ext1;
    ext_f = par.external;           /* ext_flag = # of external lines */
}

/**** RESET FLAGS ****/

inb(HB_AD);                        /* reset a/d done tecmar flags */

outb(PORTB, 0);                    /* clear port b */
inb(PORTA);                        /* reset port a */

outb(T_CONT, CLRTX | T5);          /* set T5 output high (low TC) */
outb(T_ACK, 0);                    /* clear timer interrupt */

/* PROMPT USER FOR START: START DELAY & WARNING TONES & ENABLE 8259 */

outb(T_CONT, LOADTX | T5);          /* T_DATA -> timer 5 load reg */
out2b(T_DATA, INIT_DELAY);         /* set delay in load reg */

printf("tec_init: waiting for CR to start clock, timers & first delay:");
getchar();

out(T_CONT, A_LSX | S3 | S4 | S5); /* load and arm timers */

outb(INTA01, new_mask);            /* set 8259 for new interrupt mask */

```

```

if (par.wt_dur != 0){          /* produce warning tones */
    outb(T_CONT,LOADTX | T1); /* T_DATA -> timer 1 load reg */
    temp = INIT_DELAY - par.wt_dur - par.wt_fore[0] +
        (par.s_per * par.s_stim0) / 1000 ;
    out2b(T_DATA, temp);
    outb(T_CONT,LOADTX | T2); /* T_DATA -> timer 2 load reg */
    out2b(T_DATA, par.wt_pers[par.itrial]);
    outb(T_CONT, A_LSX | S1 | S2); /* start warning tone generator */
}

/* SET PARAMETERS FOR INITIAL ENTRY INTO START SAMPLING ROUTINE */

par.reject = 1;          /* reject last trial since this is first */
par.itrial = 1;          /* par.itrial will be decremented on reject */

return;
}

```

```

/*****

INCLUDE FILE - TECMAR.H

*****/

CONTAINS ALL NECESSARY DEFINITIONS FOR TECMAR BOARD

IMPLEMENTED AS AN I/O DEVICE AT 0x0710 - 0x071F

R/W AT THE END OF THE ADDRESS COMMENTS REFER TO READ & WRITE

*****/

#define TECMAR 0x0710 /* TECMAR STARTING ADDRESS */

/*****

ADDRESS OF ALL INTERNAL TECMAR REGISTERS

*****/

/* DIGITAL / ANALOG CONVERTERS */

#define LB_0DA 0x0710 /* LBYTE OF D/A # 0 Write */
#define HB_0DA 0x0711 /* HBYTE OF D/A # 0 W */
#define LB_1DA 0x0712 /* LBYTE OF D/A # 1 W */
#define HB_1DA 0x0713 /* HBYTE OF D/A # 1 W */

/* GENERAL CONTROL */

#define CONTROL 0x0714 /* CONTROL BYTE W */
#define STATUS 0x0714 /* STATUS BYTE Read */

/* ANALOG / DIGITAL CONVERTERS */

#define CHAN_AD 0x0715 /* A/D INPUT CHANNEL # W */
#define CONV_AD 0x0716 /* START A/D CONVERSION W */
#define LB_AD 0x0715 /* LOW BYTE OF A/D R */
#define HB_AD 0x0716 /* HIGH BYTE OF A/D R */

/* TIMER CONTROLS */

#define T_ACK 0x0717 /* TIMER INT ACKNOWLEDGE W */
#define T_DATA 0x0718 /* TIMER DATA R/W */
#define T_CONT 0x0719 /* TIMER CONTROL BYTE W */
#define T_STAT 0x0719 /* TIMER STATUS BYTE R */

/* PARALLEL PORT CONTROLS */

```

```

#define PORTA    0x071C    /* PORT A          R/W */
#define PORTB    0x071D    /* PORT B          R/W */
#define PORTC    0x071E    /* PORT C          R/W */
#define P_CONT   0x071F    /* PORT CONTROL BYTE W */

```

```

/*****

```

USEFULL VALUES FOR ABOVE REGISTERS

```

*****/

```

```

/* TECMAR CONTROL - MAY BE | TOGETHER */

```

```

#define DISAUTO 0x80 /* CONTROL: disable auto increment option */
#define AD_INT   0x40 /* CONTROL: enable interrupt by a/d done */
#define OVRN_INT 0x20 /* CONTROL: enable interrupt by a/d overrun */
#define TIME_INT 0x10 /* CONTROL: enable interrupt by timer */
#define PORT_INT 0x08 /* CONTROL: enable interrupt by parallel port */
#define EXTCONV  0x04 /* CONTROL: enable external start a/d conversion */

```

```

/* TIMER CONTROLS - GENERAL */

```

```

#define MM_REG    0x17 /* T_CONT: T_DATA -> master mode register (MM) */
#define ALARM1    0x07 /* T_CONT: T_DATA -> alarm 1 register */
#define ALARM2    0x0f /* T_CONT: T_DATA -> alarm 2 register */

#define T1        0x01 /* T_CONT: T_DATA -> timer 1 mode register */
#define T2        0x02 /* T_CONT: T_DATA -> timer 2 mode register */
#define T3        0x03 /* T_CONT: T_DATA -> timer 3 mode register */
#define T4        0x04 /* T_CONT: T_DATA -> timer 4 mode register */
#define T5        0x05 /* T_CONT: T_DATA -> timer 5 mode register */
#define LOADTX    0x08 /* T_CONT= LOADTX; Tx; T_DATA -> x load reg */
#define HOLDTX    0x10 /* T_CONT= HOLDTX; Tx; T_DATA -> x hold reg */
#define H&CTX     0x18 /* T_CONT= H&CTX; Tx; T_DATA -> x hold reg & cycle */
#define SETTX0xe8 /* T_CONT= SETTX; Tx; set output of timer x high */
#define CLRTX     0xe0 /* T_CONT= CLRTX; Tx; set output of timer x low */
#define STEPTX    0xf0 /* T_CONT= STEPTX; Tx; increment (or dec) timer x */

#define S1        0x01 /* select timer 1 */
#define S2        0x02 /* select timer 2 */
#define S3        0x04 /* select timer 3 */
#define S4        0x08 /* select timer 4 */
#define S5        0x10 /* select timer 5 */
#define ARMSX     0x20 /* T_CONT= ARMS; Sx; arm timer x */
#define LOADSX    0x40 /* T_CONT= LOADSX; Sx; load timer x from x load reg */
#define A_LSX     0x80 /* T_CONT= A&LSX; Sx; arm & load timer x */

```



```

#define DISASX      0xc0  /* T_CONT= DISASX | Sx; disarm timer x from counting */
#define SAVESX      0xa0  /* T_CONT= SAVESX | Sx; save timer x count in hold reg x */
#define D&SVSX      0x80  /* T_CONT= D&SVSX | Sx; disarm & save count of timer x */

#define DDPS 0xe8  /* T_CONT: disable data pointer sequencing (bit MM14) */
#define EDPS 0xe0  /* T_CONT: enable data pointer sequencing (bit MM14) */
#define E16BDB 0xef  /* T_CONT: enable a 16 bit data bus (bit MM13) */
#define E8BDB 0xe7  /* T_CONT: enable a 8 bit data bus (bit MM13) */
#define ONFOUT 0xe6  /* T_CONT: gate on FOUT (bit MM12) */
#define OFFFOUT 0xee  /* T_CONT: gate off FOUT (bit MM12) */

#define TIM_CLR 0xff /* T_CONT: master reset to clear timer regs */

```

```

/*      TIMER CONTROLS - THIS APPLICATION      */

```

```

/**** MASTER MODE SPECIFICATIONS      ***/

```

```

#define MM_SET      0x9000 /* T_DATA: compare 1,2 disabled; TOD disabled */
                        /*      enable data pointer inc, FOUT off, BCD */
#define F1 ((long) 1000 * 1000) /* F1 system clock is 1 MHz */
#define F2 ((long) 1000 * 100) /* F2 system clock is 100 KHz */
#define F3 ((long) 1000 * 10) /* F3 system clock is 10 KHz */
#define F4 1000 /* F4 system clock is 1 KHz */
#define F5 100 /* F5 system clock is 100 Hz */

```

```

/***** TIMER 1 SPECIFICATIONS *****/

```

```

/*      USED TO GENERATE GATING FOR WARNING TONE */
/*      SET LOAD REG 1 to DELAY - WT_FORE - WT_DUR */
/*      SET HOLD REG 1 TO WT_DUR in msec */

```

```

#define TM1 0x0e42 /* T_DATA: timer 1 = disable special gate, reload */
                  /*      from hold & load reg, count once, binary */
                  /*      count down, TC toggled (set low initially) */
                  /*      no gating, count on rising edge F4= 1KHz */

```

```

/***** TIMER 2 SPECIFICATIONS *****/

```

```

/*      OUTPUT WILL BE WARNING TONE */
/*      SET LOAD REG 2 TO 1/2 WARNING TONE PERIOD */

```

```

#define TM2 0x2b22 /* T_DATA: tim 2 = disable special gate, reload from */
                  /*      from load reg, count rep, binary, count */
                  /*      down, TC toggled (for sq. wave out) */
                  /*      gated from TC of timer 1, count rising F1 */

```

```

/**** TIMER 3 SPECIFICATIONS *****/

```

```

/*      USED AS COUNTER WITH TIMER 4 TO KEEP TIME */
/*      COUNTS F3 (10KHz = 100 usec period) */

```

```

#define TM3 0x0d29 /* T_DATA: timer 3 == no gating , rising      edge of F3 */
                  /*      no special gate, reload from load, binary */
                  /*      count up rep, reload from load, high TC */

/**** TIMER 4 SPECIFICATIONS *****/
/*      TRIGGERED BY TIMER 3 TO KEEP TIME */
/*      COUNTS TC OF TIMER 3 TO MAKE 32 BIT COUNTER */

#define TM4 0x002a /* T_DATA: timer 4 == no gating , rising      edge of TC 3 */
                  /*      no special gate, reload from load, binary */
                  /*      count up rep, reload from load, toggle TC */

/**** TIMER 5 SPECIFICATIONS *****/
/*      USED TO GENERATE INTERRUPTS AFTER A FIXED DELAY */
/*      USES F1 (TM5us == 1MHz clock == micro second counter) */
/*      or F4 (TM5ms == 1KHz clock == milli second counter) */
/*      load reg 5 <- amount of delay (us or ms) */
/*      pin 2 to pin 3 on Tecmar J7 for timer 5 -> interrupts */

#define TM5us 0x0b25 /* T_DATA: timer 5 == disable special gate, reload */
                    /*      from load reg, count rep, binary, down */
                    /*      low terminal count (TC) pulse */
                    /*      no gating, rising edge of F1 (1 MHz) */
#define TM5ms 0x0e25 /*      same as above but uses F4 (1KHz clock) */

/*      PORT CONTROLS - GENERAL */

#define PA_0M 0x80 /* P_CONT: port A set for mode 0 operation */
#define PA_1M 0xa0 /* P_CONT: port A set for mode 1 operation */
#define PA_2M 0xc0 /* P_CONT: port A set for mode 2 operation */
/*      the following 6 options may be or'd to above */
/*      eg: P_CONT <- PA_1M | PA_IN | PB_1M */
#define PA_IN 0x10 /*      port A set for input (default = output) */
#define PB_1M 0x04 /*      port B set for mode 1 (default = mode 0) */
#define PB_IN 0x02 /*      port B set for input (default = output) */
#define PCU_IN 0x08 /*      port C upper nybble = input (default = out */
#define PCL_IN 0x01 /*      port C lower nybble = input (default = out */

#define PC_SET 0x01 /* P_CONT: set bit 0 in port c to 1 */
#define PC_RSET 0x00 /* P_CONT: reset bit 0 in port c to 0 */
/*      bit # is selected by bits d3,d2,d1 in P_CONT */
/*      ex: P_CONT <- PC_SET | INTE_AIN to enable int */
#define INTE_AIN 0x08 /*      set/rset PORTC bit 4 - input PORTA interrupt */
#define INTE_AOUT 0x0c /*      set/rset PORTC bit 6 - output PORTA interrupt */
#define INTE_B 0x04 /*      set/rset PORTC bit 2 - PORTB interrupts */

/*      PORT CONTROLS - THIS APPLICATION */

```

```

#define PORT_SET PA_1M | PA_IN | PB_1M | PCU_IN
                /* mode 1 for ports A & B */
                /* A = input, B = output, C upper 2 bits = in */
                /* this corresponds to #5 in tecmar 8255 manual p 6 */

/*      INTERRUPT VECTORING - THIS APPLICATION */

#define TM_IRQ   5 /* timer interrupt on IRQ5 - low memory = 34 */
#define PORTA_IRQ 4 /* port A interrupt on IRQ4 - low memory = 30 */
#define PORTB_IRQ 3 /* port B interrupt on IRQ3 - low memory = 2C */
#define SYS_CLK   0 /* system clock to be masked off to avoid interrupts */
                /* see page 5-5 in IBM/AT Technical Reference */

#define INTA00    0x20 /* 1st 8259 int controller port 1 - command byte */
#define INTA010x21 /* 1st 8259 int controller port 2 - int mask */
#define EOI       0x20 /* end of interrupt command - send to INTA00 */

```

```

;*****
title    TIMER INTERRUPT HANDLER - tm_int
;*****

;*****

include param.asm
        extrn    ic:far

codeseg segment para    public    'code'
        assume cs:codeseg
        public tm_int_

tm_int_ proc    far                ;entry point for interrupt (tm_int in C)

        pushreg                    ;save ax, di, ds, dx, es
        mov     al,EOI              ;acknowledge interrupt to 8259
        out     INTA00, al
        sti                     ;interrupts back on
        mov     ax, seg ic          ;ds points to data segment
        mov     ds, ax
        test    ic.flags, F_AD      ;check F_AD flag bit if sampling
        jnz     sample              ;

        or      ic.flags, F_DELAY    ;no sampling - delay only
        mov     dx, T_CONT           ;set delay finished flag
        mov     al, DISAS5           ;disarm timer
        out     dx, al
        mov     al, CLRT5            ;set timer 5 to clear = high (low TC)
        out     dx, al
        mov     al, STEPT5           ;step timer to force immediate clear
        out     dx, al
        jmp     tm_int_exit          ;go to exit routine

sample:                                ;sampling so set other registers

        les     di, dword ptr ic.buf_off ; set es:di -> buffer
        cld                     ;set direction for insb command
        mov     ax, 0             ;set channel number to 0
        mov     dx, CHAN_AD        ;dx points to channel register
        out     dx, al
        inc     dx                 ;dx point to start conv reg
        out     dx, al             ;start conv of chan 0

adc:     mov     dx, STATUS         ;dx points to status register
stat:    in      al, dx             ;get status register

```

```

        test    al, 80H        ;check if a/d done bit set
        jz      stat          ;not done yet

store:
        and     al, 0FH        ;al = channel number converted
        inc     dx             ;dx = lbyte register of a/d
        insb    ;es:di = lbyte, di++
        inc     dx             ;dx = hbyte register of a/d
        insb    ;es:di = hbyte, di++
        inc     al             ;al = next channel number to convert
        cmp     al, ic.nchan   ;
        je      cdone         ;all channels done ?

        dec     dx             ;dx points to channel number reg
        out     dx, al
        inc     dx
        out     dx, al        ;

        jmp     short adc      ;start conv of next chan

cdone:
        mov     ic.buf_off, di ;all channels have been converted
        mov     ax, ic.s_cur   ;save offset of next free byte in buffer
        inc     ax             ;ax = previous sample number
        mov     ic.s_cur, ax   ;increment & store new sample number
        cmp     ax, ic.s_stim0
        je      s0_on         ;time to turn stimulus 0 on ??
        cmp     ax, ic.s_stim1
        je      s1_on         ;time to turn stimulus 1 on ??
        cmp     ax, ic.s_total
        jne     tm_int_exit    ;sampling not done so exit & wait for next int

s_done:
        or      ic.flags, F_SAMP ;set sampling done flag
        mov     dx, T_CONT     ;disarm timer
        mov     al, DISAS5
        out     dx, al
        mov     al, 0          ;clear stimulus
        jmp     short stimulus

s0_on:
        mov     di, STIME0     ;di <- offset of STIME ptr in resp stru
        gettime ic.res_off[di] ;save time stim0 is turned on
        mov     al, ic.stim0    ;al = stimulus 0
        jmp     short stimulus

s1_on:
        mov     dx, PORTB      ;dx -> output port
        in      al, dx          ;get current stimulus
        or      al, ic.stim1    ;new stim = old | stimulus 1

```

```

stimulus:
    mov     dx, PORTB      ;dx points to parallel port a
    out     dx,al          ;output new stimulus

tm_int_exit:
    mov     dx, T_ACK      ;acknowledge interrupt to tecmar
    out     dx,al
    popreg
    iret                  ;restore ax, di, ds, dx, es

tm_int_endp

    codeseg ends

end

```

/*.....

SUBROUTINE - WRITEPAR(fp)

PURPOSE:

Write the parameter struct to the output parameter file

INPUT:

fp - file pointer to output parameter file

external - par: parameter structure defined in param.h

OUTPUT:

file pointed to by fp

DEPENDENCIES:

stdio.h

param.h

*****/

#include <stdio.h>

#include "param.h"

void writepar(fp)

FILE *fp; /* parameter file pointer */

{
extern PAR_STRUCT par; /* parameter structure */
int i; /* temporary index */

fprintf(fp, "%d 49d 49d 0, par.nchan, par.response, par.external);

fprintf(fp, "%d 49d 49d 0, par.ntrial, par.s_per, par.wt_dur);

fprintf(fp, "%d 49d 0, par.s_total, par.s_stim0);

for(i=0; i< par.ntrial; i++)

fprintf(fp, "%x %x %d %d %d %d0,
par.stims[i] & 0x00ff, (par.stims[i] >> 8) & 0x00ff, par.s_isi[i],
par.delays[i], par.wt_pers[i], par.wt_fore[i]);

return;

}

/*****

SUBROUTINE - WRITEPAR(fp)

PURPOSE:

Write the parameter struct to the output parameter file

INPUT:

fp - file pointer to output parameter file

external - par: parameter structure defined in param.h

OUTPUT:

file pointed to by fp

DEPENDENCIES:

stdio.h

param.h

*****/

#include <stdio.h>

#include "param.h"

void writepar(fp)

FILE *fp; /* parameter file pointer */

{
extern PAR_STRUCT par; /* parameter structure */
int i; /* temporary index */

fprintf(fp, "%d 49d 49d 0, par.nchan, par.response, par.external);

fprintf(fp, "%d 49d 49d 0, par.ntrial, par.s_per, par.wt_dur);

fprintf(fp, "%d 49d 0, par.s_total, par.s_stim0);

for(i=0; i< par.ntrial; i++)

fprintf(fp, "%x %x %d %d %d %d0,

par.stims[i] & 0x00ff, (par.stims[i] >> 8) & 0x00ff, par.s_isi[i],
par.delays[i], par.wt_pers[i], par.wt_fore[i]);

return;

}

END

FEB.

1988

DTic